الجامعة الافتراضيّة السوريّة

SYRIAN VIRTUAL UNIVERSITY

المحاكاة والنمذجة

الدكتور رند القوتلي

Books

المحاكاة والنمذجة

الدكتور رند القوتلي

من منشورات الجامعة الافتراضية السورية

الجمهورية العربية السورية 2018

Simulation and Modeling

Rand Al Kouatly

Publications of the Syrian Virtual University (SVU)

Syrian Arab Republic, 2018

الفهرس

المدخل إلى المحاكاة.. 1
ما المحاكاة؟.. 1
النظام، النموذج، والمحاكاة 2
نمذجة الحدث المتقطع Discrete Event Simulation 2
تقنية زيادة الزمن Time Advance Mechanism 4
مكونات نمذجة الحدث المتقطع Components of Discrete Event Simulation 7
بناء نموذج محاكاة.. 7
منهجية المحاكاة... 9
ملحقات.. 15
التمارين... 21
مسائل 22
المدخل إلى الاحتمالات والإحصاء 23
مقدمة.. 23
المتحولات العشوائية وخصائصها... 24
تابع توزع الكثافة الاحتمال Probability Density Function 24
توليد الأعداد العشوائية Generating Random Numbers 25
الأعداد العشوائية الزائفة Pseudo Random Numbers.................... 25
الطرق التطابقية Congruential Methods 26
طرق أخرى 26
الطريقة التطابقية العامة General Congruential Method 27
المولد التركيبي Composite Generator.................................. 27
مولد توسورث Tausworthe Generator................................. 28
طرق اختبار الأعداد العشوائية الزائفة 28
اختبارات التشغيل Runs Tests.. 29
اختبار Chi Square 30

ملحقات..31
تمارين...35
مسائل 36
توليد الأعداد العشوائية 37
مقدمة..37
طريقة التحويل العكسي The Inverse Transform Method................37
توليد الأعداد العشوائية من توابع الاحتمالية المستمرة 38
توليد الأعداد العشوائية من تابع توزع منتظم 39
توليد الأعداد العشوائية من تابع توزع أسي...................................40
توليد الأعداد العشوائية من تابع إيرلانغ Erlang...............................41
توليد الأعداد العشوائية من تابع توزع الطبيعي...............................42
توليد الأعداد العشوائية من توابع الاحتمالية المتقطعة.........................43
توليد الأعداد العشوائية من تابع توزع الهندسي...............................44
توليد الأعداد العشوائية من تابع توزع الثنائي..................................45
توليد الأعداد العشوائية من تابع توزع بويسن..................................45
توليد الأعداد العشوائية من توابع الاحتمالية التجريبية 46
توليد الأعداد العشوائية التجريبية من توابع الاحتمالية المتقطعة 46
توليد الأعداد العشوائية التجريبية من توابع الاحتمالية المستمرة................48
طريقة الرفض Rejection Methods 50
طريقة مونتيكارلو Monte Carlo Method 52
ملحقات..53
مسائل 55
تصميم المحاكاة..56
مقدمة..56
طريقة تقدم الحدث Event Advance Design..................................56
قائمة الأحداث المستقبلية Future Event List..................................58
طريقة المصفوفة التسلسلية Sequential Array Method 59

طريقة القوائم المترابطة Linked Lists Method............................60
تحقيق القوائم المترابطة Implementation Of Linked Lists 63
القوائم المترابطة الزوجية Double Linked Lists 67
طريقة تقدم واحدة الزمن Unit Time Advanced 67
اختيار واحدة الزمن Selecting Unit Time 71
استخدامات طريقة تقدم واحدة الزمن، وطريقة تقدم الحدث 71
طريقة محاكاة النشاط Activity Based Simulation Design 72
أمثلة Examples 75
نظام جرد مستودع Inventory System 75
نظام الصف الدائري Round Robin Queue.............................. 80
ملحقات... 85
التمارين.. 88
مسائل 89
تقنيات التقييم لمخرجات نظم المحاكاة... 90
مقدمة.. 90
تحصيل المخرجات Collecting endogenously created data......... 90
الحالات المستقرة والعابرة Transient states steady states simulation
.... 91
الحالات العابرة في المحاكاة Transient states simulation 92
الحالات المستقرة في المحاكاة Steady state simulation 92
تقنيات تقييم الحالات المستقرةEstimating techniques for SS.......... 93
تقييم مجال الثقة لمتوسط المتحول العشوائي.. 93
التقييمات الأخرى للمتحول العشوائي... 102
طرق تقييم الحالات العابرة Estimation techniques for transient state
.... 105
التجارب الأولية Pilot experiments.. 106
ملحقات.. 110

مسائل 161
لغات المحاكاة.. 162
مقدمة.. 163
مقارنة بين الحزم البرمجية ولغات البرمجة 164
تصنيف الحزم البرمجية... 166
مقارنة بين الحزم البرمجية لأغراض عامة أو لتطبيقات خاصة............... 167
العوامل المشتركة بين الحزم البرمجية 168
مواصفات لغات البرمجة... 169
الحزم البرمجية عامة التوجه General purpose Simulation packages175
حزمة Arena.. 176
حزمة Extendsim 184
الحزم البرمجية غرضية التوجه Application oriented simulation packeges.. 189
ملحقات.. 190

المدخل إلى المحاكاة

Introduction to Simulation

1. ما المحاكاة؟

- تهدف المحاكاة باستخدام الحاسب إلى محاكاة Simulate أو تقليد Imitate نظم Systems موجودة ومستخدمة في الواقع.
- لدراسة هذه النظم يجب وضع فرضيات حول طريقة عملها، هذه الفرضيات تؤدي عادة إلى الحصول على معادلات رياضية ومنطقية. تشكل هذه المعادلات والفرضيات نموذج النظام Model of the system.
- عندما تكون العلاقات التي تشكل النموذج قليلة التعقيد، بحيث يمكن للمعادلات الرياضية أن تعبر بشكل تام عن النظام يدعى النموذج عندها بالنموذج التحليلي Analytical Model.
- عندما يكون النظام معقد (كما في الحياة العملية)، يمكن عندها دراسة النظام فقط باستخدام المحاكاة Simulation.
- تستخدم المحاكاة Simulation باستخدام الحواسب الالكترونية لجمع المعلومات والمعطيات وذلك لتوقع طبيعة النظام الحقيقية. [1 إيضاح]

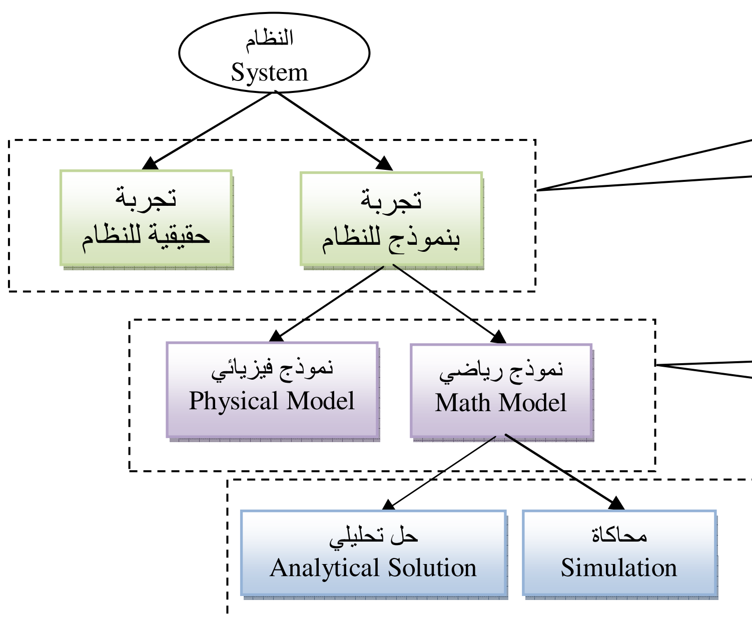
عند وضع الفأرة على المحاكاة [1 إيضاح]

عند وضع الفأرة على مكونات [2 إيضاح]

عند وضع الفأرة على المكونات [2 إيضاح]

2. النظام، النموذج، المحاكاة

- يمكن أن يعرف النظام System بأنه عبارة عن مجموعة مكونات تتفاعل مع بعضها البعض لتكون الهدف منها.
- في الحياة العملية يتعلق وصف النظام بالهدف الذي يدرس من اجله، فمثلا يمكن أن تشكل مجموعة المكونات نظاماً من اجل دراسة معينة، و يمكن أن يشكل نظاماً جزئياً Subsystem لنظام أكبر. [1 مثال]
- يمكن تقسيم النظام إلى صنفين، مستمر Continuous و متقطع Discrete:
  - في النظام المتقطع تتغير معاملات النظام عند فترات زمنية متفاوتة وغير ثابتة. [2 مثال].
  - في النظام المستمر تتغير معاملات النظام بشكل مستمر مع الزمن. [3 مثال].
- في معظم الأنظمة فانه من الضروري تجربة النظام من اجل الحصول على رؤية معمقة عن معاملات النظام، كما يمثل الشكل-1.



عند وضع الماوس فوق المنطقة المنقطة يظهر الإيضاح 3.

عند وضع الماوس فوق المنطقة المنقطة يظهر الإيضاح 4.

عند وضع الماوس فوق المنطقة المنقطة يظهر الإيضاح 5.

الشكل -1: لماذا ندرس النظام

للقيام بمحاكاة نظام ما باستخدام النموذج الرياضي، يجب استخدام وسائل خاصة لتحقيق ذلك، إذا من المفيد تصنيف نماذج المحاكاة إلى ثلاثة أصناف:

- **النموذج الستاتيكي Static أو الديناميكي Dynamic:** نموذج المحاكاة الستاتيكي هو تعبير عن نظام ما عند وقت محدد [4 مثال]. أما النموذج الديناميكي فهو تعبير عن نظام ما خلال فترات زمنية محددة. [5 مثال]
- **النموذج المحدد Deterministic أو العشوائي Stochastic:** النموذج المحدد هو النموذج الذي لا يحتوي على أي محددات عشوائية [6 مثال]. أما النموذج العشوائي فهو النموذج الذي يحتوي على الأقل محدد واحد عشوائي. [7 مثال]
- **النموذج المستمر والمتقطع:** في النظام المتقطع تتغير معاملات النظام عند فترات زمنية متفاوتة وغير ثابتة. بينما في النظام المستمر تتغير معاملات النظام بشكل مستمر مع الزمن، كما بينا سابقا.

3. نمذجة الحدث المتقطع Discrete Time Simulation

- إن نمذجة الحدث المتقطع هو نمذجة النظام بالنسبة لتغيراته بالزمن، حيث يتم تمثيل التغيرات الدائمة لمتحولات النظام بالنسبة للزمن، [6 إيضاح].
- تعرف التغيرات في متحولات النظام التي تحدث عند نقاط محددة بالزمن بالحدث event.
- بالرغم انه يمكن نمذجة نظام الحدث المتقطع يدوياً إلا انه في الحياة العملية لا يمكن التعامل مع كمية المعلومات الضخمة التي يتعامل معها النظام لذلك تستخدم الحواسب الالكترونية. [8 مثال]

1.3. تقنية زيادة الزمن Time- Advance Mechanism

- بسبب الطبيعة الديناميكية لنمذجة الحدث المتقطع فانه من الضروري تعقب بدقة تغيرات الزمن المستخدم في المحاكاة، وكذلك يجب تحديد آلية لزيادة الزمن خلال التغيرات من حدث إلى حدث آخر.
- يدعى المتحول المستخدم لنمذجة تغيرات الزمن في نظام محاكاة ساعة المحاكي Simulation Clock.
- إن وحدة الزمن المستخدمة لساعة المحاكي غير محددة تماماً وذلك عند تصميم برنامج المحاكاة بلغات برمجة عالية المستوى [7 إيضاح]، ويمكن اعتبار وحدة الزمن مساوية إلى تغيرات مدخلات النظام. [8 إيضاح]
- توجد طريقتان لتحديد آلية لزيادة الزمن خلال المحاكاة:
  - آلية الحدث القادم Next even time Advanced: وتستخدم في معظم نظم المحاكاة، حيث يزاد ساعة المحاكي عند ظهور حدث جديد.
  - آلية زيادة الزمن الثابت Fixed increment time advance: حيث يزاد متحول ساعة المحاكي بصورة ثابتة مع مرور الزمن. وتعتبر حالة خاصة من النوع الأول.

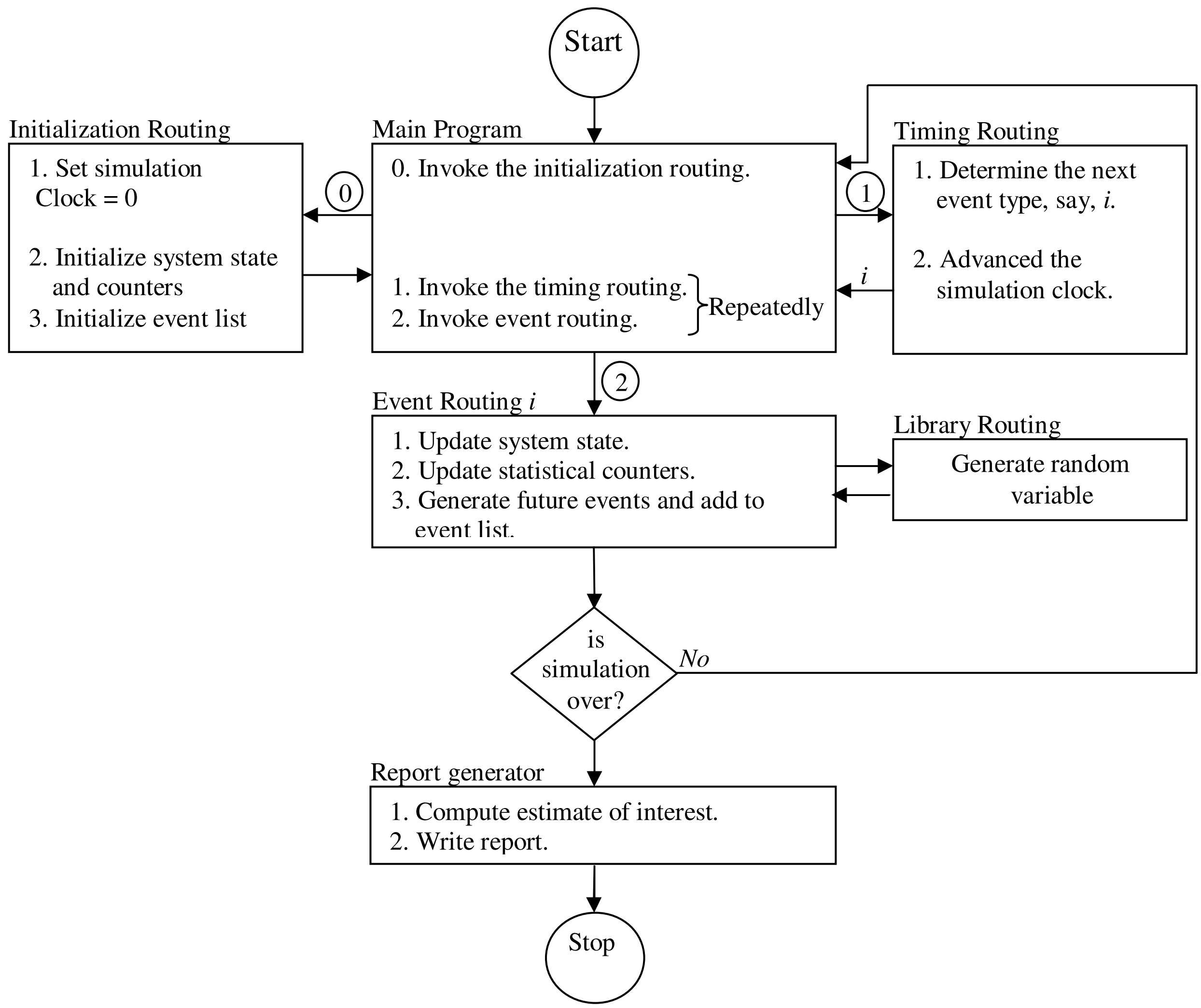
- عند استخدام آلية الحدث القادم يتم في البداية جعل قيمة متحول ساعة المحاكي مساوياً إلى الصفر، ويتم بعد ذلك تحديد الأزمان المتوقعة للأحداث القادمة.
- عند ظهور أول حدث يتم تحديث متحولات النظام ويتم تحديد قيمة ساعة المحاكي عند قيمة أول زمن بدقة، يتم كذلك تحديث قيم الأزمنة المتوقعة للأحداث القادمة.
- عند ظهور الحدث التالي، يتم أيضا تحديث متحولات النظام ويتم تحديد قيمة ساعة المحاكي عند قيمة ثاني زمن بدقة، يتم كذلك تحديث قيم الأزمنة المتوقعة للأحداث القادمة، وهكذا تستمر العملية حتى ظهور شرط التوقف.

[9 مثال]

2.3. مكونات نمذجة الحدث المتقطع Components of Discrete Event Simulation

- بالرغم من تنوع نظم المحاكاة باستخدام نمذجة الحدث المتقطع إلا أنها تشترك جميعها بعدد من المكونات، بحيث يمكن أن تنظم هذه المكونات لتساعد المبرمجين في تعديل البرمجيات لاحقاً وكشف الأخطاء.
  - حالة النظام System state: ويمثل مجموعة المتحولات المستخدمة في معرفة حالة النظام عند زمن محدد.
  - ساعة المحاكي Simulation Clock: ويمثل متحول يحتفظ بالقيمة الحالية للزمن في المحاكاة.
  - قائمة الحدث Event list: وهي قائمة تحتوي على طبيعة الحدث القادم.
  - عدادات إحصائية Statistical counters: وهي عبارة عن متحولات تحفظ قيم الإحصائية عن أداء النظام.
  - برنامج ابتدائي Initialization routine: برنامج جزئي يستخدم لإعطاء القيم الابتدائية لجميع المتحولات عند البدء بالمحاكاة (أي عند الزمن 0).
  - برنامج التوقيت Timing routine: برنامج جزئي يستخدم قائمة الحدث لمعرفة الحدث القادم ومن ثم تحديث قيمة متحول ساعة المحاكي لتصبح مساوية لزمن حدوث الحدث الجديد.
  - برنامج الحدث Event routine: برنامج جزئي يحدث قيم متحولات النظام عند قدوم حدث ما (في أي نظام محاكاة يوجد برنامج جزئي واحد لكل حدث).
  - برامج المكتبة Library routines: وهي عبارة عن مجموعة برامج جزئية تقوم بتوليد قيم عشوائية المطلوبة لنظام المحاكاة.
  - مولد التقارير Report generator: وهو عن برنامج جزئي يستخدم العدادات الإحصائية لكتابة التقارير عن نظام المحاكاة.
  - البرنامج الرئيسي Main program: وهو البرنامج الجزئي المسؤول عن إدارة نظام المحاكاة، ومن ثم إظهار التقارير وإنهاء المحاكاة.

o الشكل التالي يبين المخطط الانسيابي Flow control لنظام نمذجة الحدث المتقطع.



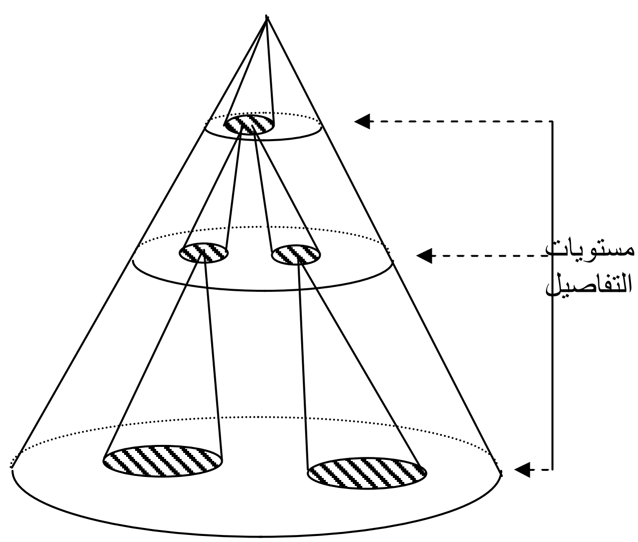
الشكل -2: نظام نمذجة الحدث المتقطع.

4. بناء نموذج محاكاة Building a simulation model

- يتصف أي نظام بالخصائص التالية:

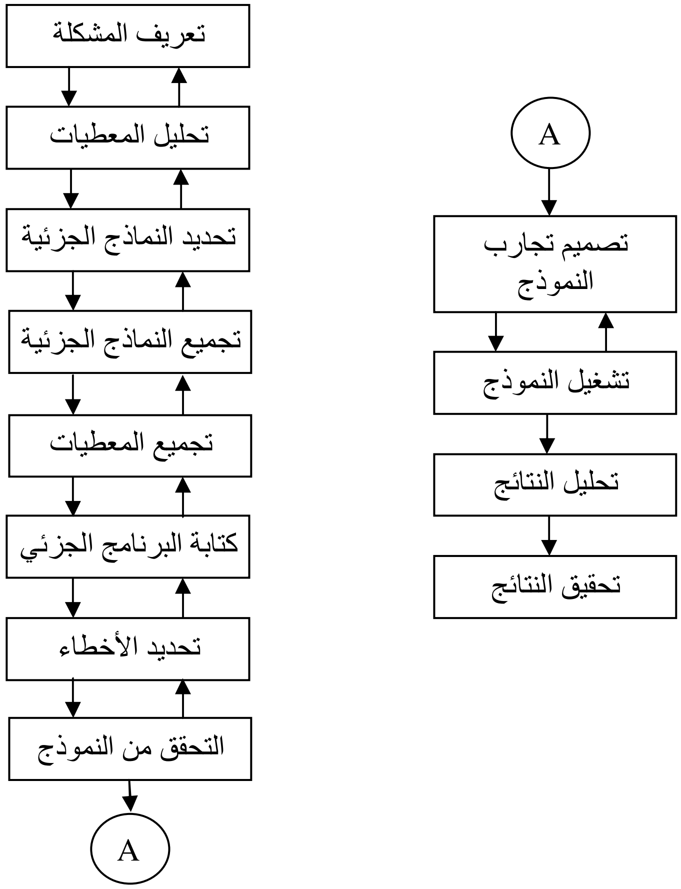
  - البيئة Environment: أي نظام يمكن إن يرى عبارة عن نظام جزئي من نظام أوسع.
  - الاعتمادية Interdependency: لا يمكن لأي فاعلية أن تكون مستقلة عن غيرها.
  - النظام الجزئي Sub-system: يمكن لأي نظام أن يقسم إلى أنظمة جزئية.
  - التنظيم Organization: كل الأنظمة عبارة عن مجموعة مكونات منظمة بشكل كبير بحيث تتفاعل مع بعضها البعض لتكون النظام.

- عند بناء نظام نمذجة لمحاكاة نظام في حقيقي، فان ذلك يتم عن طريق محاكاة نظم جزئية عند مستويات وتفاصيل مختلفة ومن ثم تجميع هذه النظم الجزئية لتكوين النظام الكلي.
- يبين الشكل التالي كيف يتم نمذجة أنظمة جزئية مختلفة عند مستويات مختلفة، لتشكل بمجموعها النظام الكامل.



الشكل -3: هرم إدارة بريد

- إن الخطوات الأساسية المتبعة لبناء مبينة في الشكل التالي:



الشكل -4: الخطوات الأساسية المتبعة لبناء نظام محاكاة

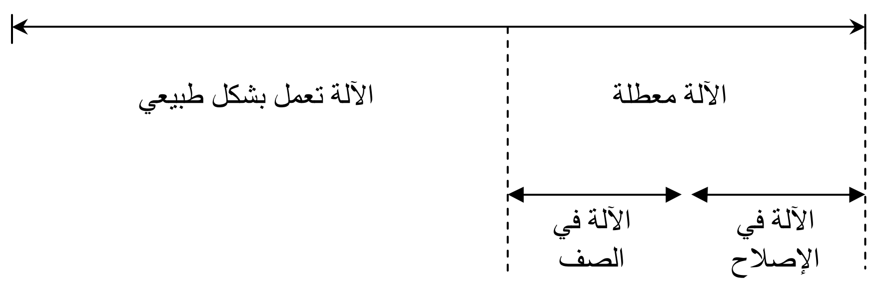
5. منهجية المحاكاة Simulation methodology

سنبين في هذه الفقرة المنهجية المتبعة في بناء نظام محاكاة عبر عرض مثال بسيط ومشهور، وهو نظام خدمة الزبون لمخدم وحيد Single Server Queuing.

المسألة:

لنفترض **أننا نريد** بناء نظام محاكاة لنظام مخدم وحيد بعدد زبائن محدود. لنفترض انه لدينا مصنع تعمل فيه آلات تساوي M، لنفترض أن كل اله تعمل لفترة زمنية محددة ثم تتعطل وتتوقف عن العمل. ولنفترض أيضا انه يوجد لدينا عامل إصلاح واحد فقط مسؤول عن إصلاح الآلات المعطلة وبحيث تبقى الآلة معطلة حتى يتم إصلاحها من قبل العامل. ويتم الإصلاح وفق مبدأ المتعطل أولاً يخدم أولاً FIFO، بحيث تبقى الآلات المعطلة تنتظر في الصف حتى يأتي دورها في الإصلاح.

وبالتالي كل آلة من الآلات المصنع سوف تتبع دورة العمل التالية:



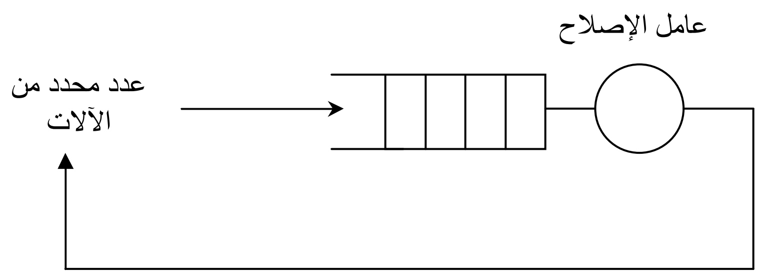
الشكل -5: دورة عمل الآلة الأساسي.

المطلوب

بناء نظام محاكاة لتحديد زمن توقف الآلات عن العمل.

الحل:

- لمعرفة زمن توقف الآلة يجب معرفة الزمن اللازم لإصلاح آلة وانتظار الآلة في الصف.
- يمكن تشبيه عمل عامل الإصلاح بنظام بمخدم وحيد بعدد زبائن محدود.



الشكل -6: مشكلة تعطل الآلة.

- لنفترض أن زمن عمل الآلة بشكل طبيعي هو عشر وحدات زمنية Unit time، ولنفترض أن زمن إصلاح الآلة أيضاً ثابت ويساوي خمس وحدات زمنية.

[10 إيضاح]

- الخطوة الأولى تتم بتحديد الأحداث events المهمة والتي يؤدي ظهورها إلى تغير أحداث في النظام.

يمكن تمثيل الأحداث بالنظام باستخدام مخططات الأحداث event-grave حيث يمثل كل حدث عبارة عن عقدة تتصل بالعقد الأخرى بواسطة أسهم تشير إلى انتقال الأحداث في النظام.

[10 مثال]

- الخطوة الثالثة تتم بتحديد متحولات التي تصف الأحداث المتغيرة للنظام.
- الخطوة الثالثة هي تحديد أداء النظام الذي يمكن أن يغير متحولات النظام السابق الذكر.

[11 إيضاح]

يمكن تحديد حدثين مهمين:

1- تعطل الآلة ودخولها في صف الانتظار. [12 إيضاح]

2- إصلاح الآلة وخروجها من صف الانتظار. [13 إيضاح]

- لتحديد بدقة الحدثين السابقين يجب إضافة مجموعة متحولات جديدة، تعرف بمتحولات الساعة، والذي يلاحق مسار تغيرات وقت حدوث وانتهاء الأحداث السابقة.
- سوف يتم استخدام متحول ساعة لكل آلة، حيث سوف يبين هذا المتحول اللحظة الزمنية التي تتعطل فيها الآلة وتدخل في صف الإصلاح.
- كما سوف يستخدم متحول ساعة آخر ليحدد الزمن التي يتم فيها إصلاح الآلة.
- إي إذا كان لدينا m آلة فانه نحتاج إلى $m+1$ متحول ساعة بحيث يستخدم m متحول ليصف زمن تعطل كل آلة من الآلات بينما نستخدم متحول واحد ليحدد زمن إصلاح أية آلة.
- كذلك يجب استخدام متحول ساعة عام master clock ليلاحق مسار متغيرات نظام المحاكاة.

لنفترض إننا نرغب بإجراء محاكاة لثلاث آلات في المعمل:

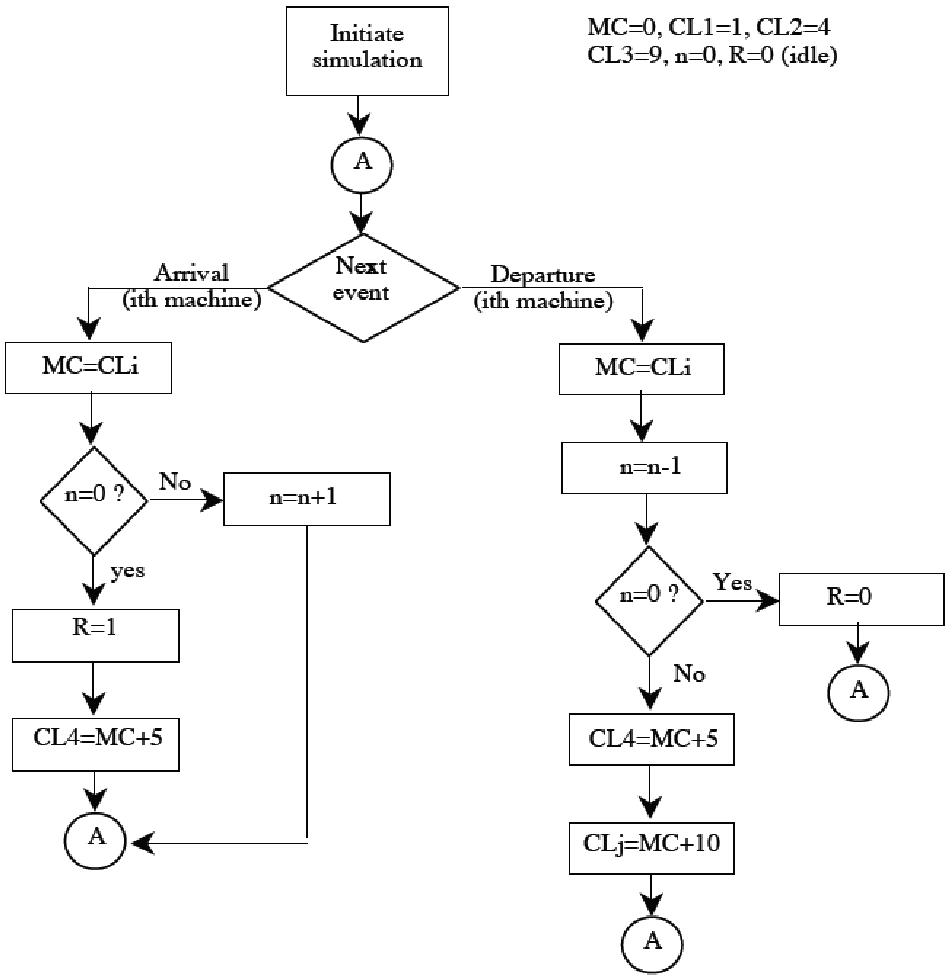
- لنفترض أن متحولات الساعة للآلات الثلاث هي CL1، CL2 و CL3 والتي تصف زمن تعطل الآلات الثلاث.
- لنفترض أن المتحول الذي يصف إصلاح الحد الآلات السابقة هو CL4.
- وان متحول الساعة العام هو MC.
- وان المتحول R يصف ما إذا كان العامل يقوم بالإصلاح أم في حالة راحة.
- لنفترض أن الشروط الابتدائية والتي يحدث خلالها الأعطال هي: CL1=1، CL2=4, و CL3=9، وان زمن إصلاح الآلة الواحدة هو خمس وحدات زمنية وان زمن عمل الآلة هو عشر وحدات زمنية كما بينا سابقا.

الجدول.١: محاكاة النظام يدوياً.

	n	CL1	CL2	CL3	CL4	MC	*حالة العامل*
1	0	1	4	9	-	0	راحة
2	1	-	4	9	6	1	مشغول
3	2	-	-	9	6	4	مشغول
4	1	16	-	9	11	6	مشغول
5	2	16	-	-	11	9	مشغول
6	1	16	21	-	16	11	مشغول
7	1	-	21	26	21	16	مشغول

[14 إيضاح]،[15 إيضاح].

يتم تحقيق محاكاة النظام السابق باستخدام لغة برمجة عالية المستوى، يبين الشكل التالي المخطط التدفق لبرنامج المحاكاة السابق.



الشكل -7: مخطط التدفق لمحاكاة نظام إصلاح الآلات في مصنع.

[16 إيضاح]

ملحقات

[1 إيضاح]

كمثال على استخدام المحاكاة، لنفترض أن مدير مصنعاً يريد يقيم دراسة لتوسيع مصنعه وذلك عن طريق بناء قسم إضافي. ولكن مجلس إدارة المصنع غير متأكد من أن التوسع سوف يعطي المردود الاقتصادي الذي يغطي تكاليف الإنشاء، فمن غير المعقول بناء التوسع ومن ثم هدمه إذا لم يغطي تكاليفه الاقتصادية. (يمكن الحصول على الايجابة باستخدام المحاكاة لدراسة المردود الاقتصادي للتوسع).

[2 إيضاح]

أشخاص أو آلات.

[1 مثال]

فمثلاً لدراسة نظام خدمة الزبائن في مصرف وذلك لتحديد عدد العاملين المفترض توظيفهم للحصول على خدمة جيدة في صرف أو تحصيل الشيكات. فالنظام هو عبارة عن نظام جزئي يدرس العلاقة بين الموظفين و الزبائن المنتظرين في الصف لتخديمهم.

[2 مثال]

نظام خدمة الزبائن في مصرف يعتبر نظام متقطع حيث يتغير عدد الزبائن في الصف عند قدوم زبون جديد أو عند الانتهاء من تخديم زبون.

[3 مثال]

نظام الطائرة يعتبر نظام مستمر حيث يتغير معاملات النظام (كارتفاع وموقع الطائرة) بشكل مستمر مع الزمن.

[3 إيضاح]

للمقارنة بين نتائج التجربة العملية للنظام ونتائج والنتائج العملية في حال تغير الشروط عندها نحصل على نتائج دقيقة للنظام. إلا ان هذه الطريقة مكلفة اقتصادياً.

[4 إيضاح]

وهي مقارنة بين نموذج فيزيائي (مثلا دراسة تأثير حركة الهواء مولد بواسطة مراوح ضمن غرفة)، وبين النموذج الرياضي المقابل.

[5 إيضاح]

لدراسة مدى وثوقية النموذج الرياضي، وذلك بتحليل النتائج الرياضية يدويا، أو باستخدام المحاكاة عندما يكون التحليل الرياضي معقدا.

[4 مثال]

نموذج مونتي كارلو Monte Carlo.

[5 مثال]

نظام التغليف في مصنع.

[6 مثال]

نظام المحاكاة لتفاعل الكيماوي.

[7 مثال]

نظام خدمة الزبائن في مصرف.

[6 إيضاح]

عمليا تتغير متحولات النظام بالنسبة لنقاط زمنية بعدد محدد تقريباً.

[8 مثال]

لنفرض إننا نريد محاكاة نظام خدمة الزبائن في متجر للحلاقة حيث يقوم حلاق واحد بخدمة الزبائن المنتظرين، وبحيث يراد حساب الزمن الذي يستغرقه الزبون في الانتظار (الفارق الزمني بين قدوم الزبون إلى المتجر والزمن الذي يبدأ فيه بالحلاقة). ويمكن تحديد متحولات النظام بـ:

- حالة الحلاق (المخدم) هل هو مشغول idle، أم غير مشغول busy. ويفيد المتحول في معرفة هل الزبون الجديد سوف يخدم مباشرتاً (عندما يكون الحلاق غير مشغول) أو ينتظر في الصف (عندما يكون الحلاق مشغولاً).
- عدد الزبائن المنتظرين في الصف. ويفيد المتحول في معرفة ما إذا كان المخدم سوف يصبح غير مشغول، أي يخدم أول زبون منتظر في الصف.
- زمن قدوم الزبون إلى متجر الحلاقة. ويفيد هذا المتحول في معرفة زمن انتظار الزبون في الصف.

يوجد حدثين لهذا النظام، وهما قدوم الزبون إلى متجر الحلاقة، والانتهاء من تخديم الزبون.

[7 إيضاح]

كلغة C مثلا.

[8 إيضاح]

في معظم الحالات لا توجد علاقة بين زمن المحاكاة وبين الزمن الحقيقي الذي يستغرقه برنامج المحاكاة أثناء تنفيذه على الحاسب.

[9 إيضاح]

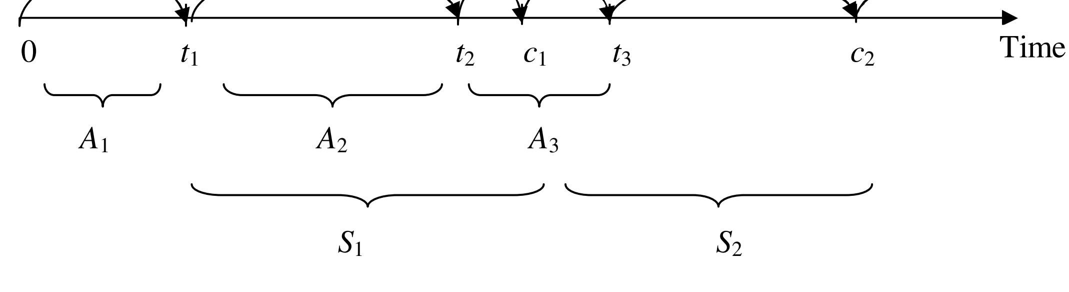
سوف نستخدم الطريقة الأولى (آلية الحدث القادم Next even time Advanced) في جميع نظم المحاكاة التي سوف نتحدث عنها في هذا المقرر.

[9 مثال]

لنحدد الآن متحولات النظام لنظام محاكاة لمتجر حلاق المبين في [٨ مثال]:

- t_i: يمثل زمن قدوم الزبون رقم i إلى محل الحلاقة.
- $A_i=t_i-t_{i-1}$: يمثل فرق الزمن بين قدوم الزبون رقم i و الزبون رقم i-1 إلى محل الحلاقة.
- S_i: الزمن الذي يستغرقه الحلاق لخدمة الزبون رقم i (بدون زمن الانتظار).
- D_i: الزمن الذي ينتظره الزبون ذو الرقم i في الصف (زمن التأخير) حتى يتم تخديمه.
- $c_i=t_i+D_i+S_i$: الزمن اللازم لانتهاء من تخديم الزبون ذي الرقم i (زمن رحيل الزبون).
- e_i: زمن ظهور الحدث i.

يبين الشكل التالي مثالا على دخول ثلاثة زبائن إلى متجر الحلاقة، يخدم الزبون الأول فور قدومه إلى متجر الحلاقة بسبب كون الحلاق غير مشغول. يأتي الزبون الثاني عند الزمن t_1 وينتظر في الصف كون الحلاق مشغول في تخديم الزبون الأول، ثم يأتي الزبون الثاني عند الزمن t_2 وينتظر في الصف كون الحلاق مشغول في تخديم الزبون الأول. عند الزمن c_1 يغادر الزبون الأول ويبدأ بتخديم الزبون الثاني فيكون زمن انتظار الزبون الثاني في الصف هو A_1، وهكذا...

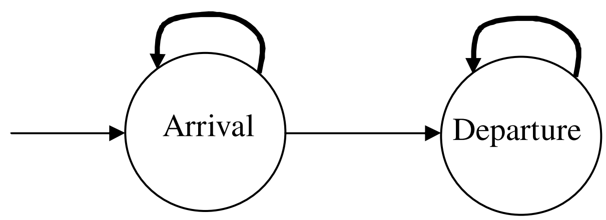


[10 إيضاح]

يمكن أن ندرس المسالة بتعقيد اكبر وذلك بافتراض ان زمن عمل وإصلاح الآلات غير متساوي وغير ثابت وإنما يتغير بشكل عوائي.

[10 مثال]

يمكن أن نمثل مخطط الأحداث لنظام محاكاة صف وفق الشكل التالي:



[11 إيضاح]

في مثالنا:

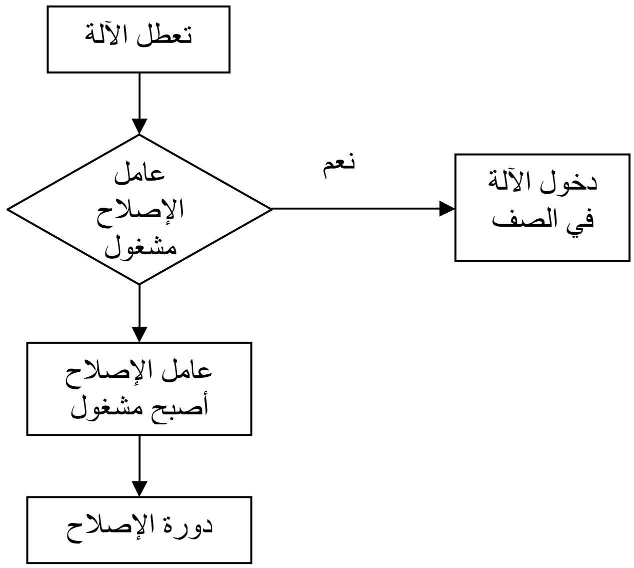
إن الحدث المهم الملاحظ في هذا النظام هو تعطل الآلة.

أما متحول النظام التي تصف الحدث السابق هو عدد الآلات المعطلة n.

- إذا كان $n=0$ فان عدد الآلات المعطلة مساويا للصفر إي لا يوجد آلات تنتظر في الصف، وعامل الإصلاح في حالة راحة.
- إذا كان $n=1$ فانه توجد آله واحدة معطلة ولا توجد الآلات تنتظر في الصف، ولكن عامل الإصلاح يعمل بإصلاح الآلة.
- في حالة $n>1$ فانه توجد n-1 اله تنتظر بالصف لحين إصلاحها وعامل الإصلاح يقوم بإصلاح الآلة الأولى المعطلة.

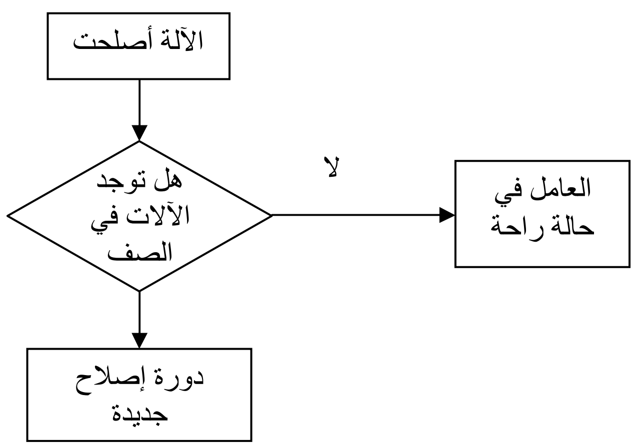
[12 إيضاح]

يوصف هذا الحدث وفق المخطط الانسيابي التالي:



[13 إيضاح]

يوصف هذا الحدث وفق المخطط الانسيابي التالي:

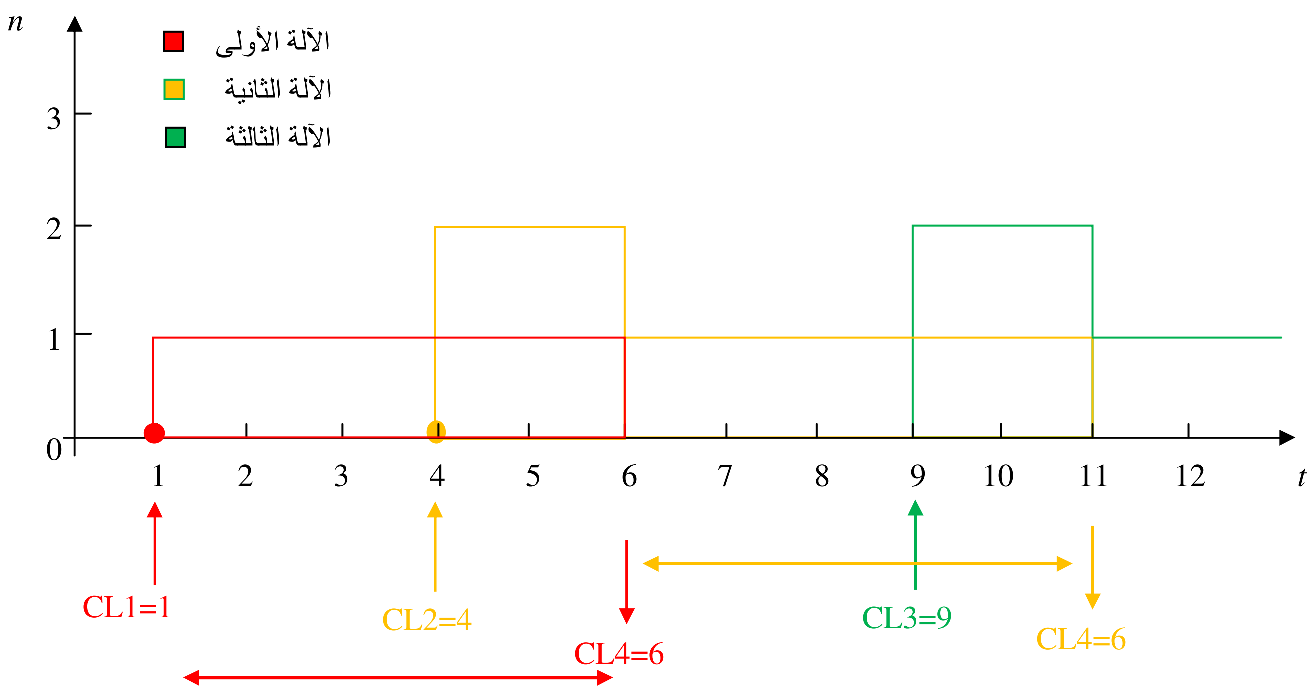


[14 إيضاح]

نلاحظ من الحل اليديوي للنظام السابق أن الأحداث تحدث خلال أزمنة ثابتة 0، 1، 4، ٦، ٩، 11، 16،..... بين هذه الأحداث يكون النظام مستقراً. أي يكفي لمراقب النظام التحقق من النظام في هذه الأوقات فقط.

[15 إيضاح]

يمكن أن يوصف النظام بطريقة أوضح وفق المخطط الزمني التالي:



[16 إيضاح]

على الطالب القيام بكتابة برنامج يحقق المخطط السابق ويقوم بتنفيذه وملاحظة النتائج عن قيم مختلفة من المتحولات والشروط الابتدائية.

التمارين

1- هل تعتقد أن نظام المحاكي في مسألة إصلاح الآلات في مصنع قد أخذ بعين الاعتبار أن زمن إصلاح الآلة متغير.

A) نعم أخذ بعين الاعتبار.

B) لا لم يأخذ بعين الاعتبار.

2- هل تعتقد أن نظام المحاكي في مسألة إصلاح الآلات في مصنع قد أخذ بعين الاعتبار أن زمن عمل الآلة ثابت.

A) نعم أخذ بعين الاعتبار

B) لا لم يأخذ بعين الاعتبار.

3- ما هو عدد عمال الإصلاح في نظام المحاكاة في مسألة إصلاح الآلات في مصنع.

A) ثلاث عمال.

B) عاملين.

C) عامل واحد.

D) متغير حسب عدد الآلات المعطلة.

4- ماهو النظام المعتمد في في إصلاح الآلات في مصنع.

A) القادم أولاً يخدم أولاً FIFO.

B) يخدم الآلة التي تحتاج إلى وقت أصغر في الإصلاح.

C) القادم أولاً يخدم أخراً LIFO.

D) بشكل عشوائي.

مسائل

1- أكتب برنامجا يقوم بإجراء محاكاة لنظام مخدم زبون وحيد لإصلاح الآلات في مصنع، بحيث يقوم البرنامج بطباعة المخرجات التالية عند ظهور كل حدث، وكرر التنفيذ حتى الوصول إلى القيمة 20 للساعة الرئيسية Master clock. ثم تأكد من أن نظام المحاكاة التي قمت بتنفيذه صحيحاً عن طريق المحاكاة اليديوية.

2- قم بتنفيذ نظام محاكاة يدوية لنظام مخدم زبون في مصرف بحيث أن عدد المخدمين المستخدمين واحد، ثم قم بكتابة البرنامج للنظام وقم بتنفيذه كما في السؤال السابق.

3- كرر السؤال السابق عندما يكون لدينا مخدمين يقومان بتخديم الزبائن.

المدخل إلى الاحتمالات و الإحصاء

Basic Probability and Statistics

1. مقدمة

- لا يكفي فقط لتحقيق أية نظام محاكاة بشكل صحيحي إلى استنتاج مخطط التدفق النظام ومن ثم تحقيق المخطط باستخدام برامج حاسوبية عالية المستوى، وإنما يحتاج نظام المحاكاة إلى مجموعة من متحولات عشوائية و إحصائية ليكون اقرب إلى الواقع.
- فمثلاً يتم استخدام الاحتمالات والإحصاءات لمحاكاة التصرف العشوائي للنظام، ومن ثم للتحقق من جودته وكفاءته،
- أو يتم استخدام الاحتمالات والإحصاءات لاختيار تابع التوزيع الاحتمالي لمتحولات الدخل، او لتوليد متحولات عشوائية تحاكي هذا التوزيع، أو لتحليل نتائج النظام و تصميم التجربة العملية الكاملة لنظام المحاكاة.

سوف نستعرض في هذا الدرس بعض الأفكار الأساسية عن الاحتمالات والأعداد العشوائية والمتعلقة بنظام المحاكاة.

2. المتحولات العشوائية وخصائصها

- تعرف التجربة experiment بأنها عملية أو إجرائية تكون نتائجها غير محددة و غير معروفة تماماً.
- تشكل مجموعة النتائج المحتملة للتجربة بالفضاء البسيط Sample Space ويرمز لها بـ S، بينما تسمى بعض نتائج التجربة بالنقاط البسيطة. [1 مثال] [2 مثال].
- يعرف المتحول العشوائي Random Variable بأنه عبارة عن تابع أو قانون مهمته تخصيص أي رقم حقيقي محصور بين -∞ و +∞ بنقطة من الفضاء البسيط S. [3 مثال]، [1 إيضاح].
- يعرف تابع توزع الاحتمالي $F(x)$ distribution function للمتحول العشوائي X وللقيم الحقيقية x كما يلي:

$$F(x) = P(X \leq x) \qquad \text{for} -\infty < x < \infty$$

حيث $P(X \leq x)$ هو احتمال مرتبط بالحدث $\{X \leq x\}$. [2 إيضاح].

- يعرف المتحول العشوائي المتقطع Discreet Random Variable بأنه عبارة عن مجموعة من الأرقام العشوائية x_1، x_2،... [4 مثال] التي تأخذ قيم محددة.
- يعطى تابع الاحتمال لمتحول العشوائي المتقطع بالعلاقة:

$$p(x_i) = P(X \le x_i) \qquad \text{for } i = 1,2,... \Rightarrow \sum_{i=1}^{\infty} p(x_i) = 1$$

- ليكن لدينا المجموعة $I=[a,b]$ حيث a و b أعداد حقيقية تحقق $a \le b$، فان:

$$P(X \in I) = \sum_{a \le x_i \le b} p(x_i)$$

حيث ∋ تدل على الاحتواء. أي احتمال أن المتحول X ينتمي إلى المجموعة I يساوي إلى مجموع احتمال $p(x_i)$ لجميع قيم x_i المحصورة ضمن المجال $a \le x_i \le b$.

- إن تابع التوزع الاحتمال للمتحول العشوائي المتقطع يعطى:

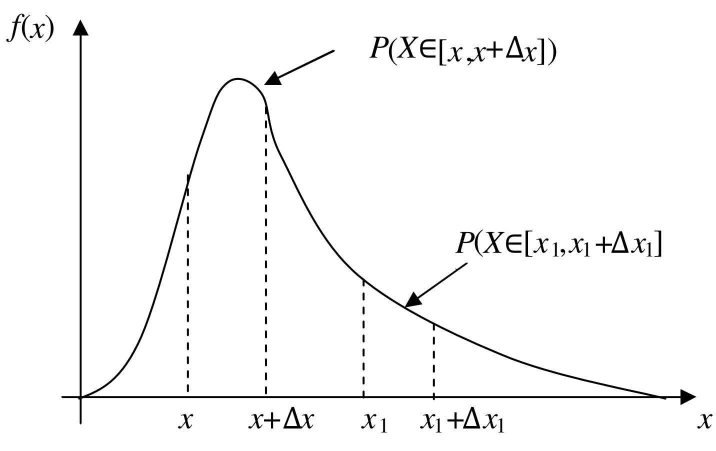
$$F(x) = \sum_{x_{i_i} \le x} p(x_i) \qquad \text{for all } -\infty < x < \infty$$

3. تابع توزع الكثافة الاحتمال Probability Density Function

- لنفترض انه لدينا متحول عشوائي يمكن أن يأخذ عدد لانهائي من القيم [5 مثال]، يقال عن العدد العشوائي X المنتمي للمجموعة السابقة بأنه مستمر Continuous إذا وجد تابع $f(x)$ ولأي رقم حقيقي B يحقق:

$$P(X \in B) = \int_B f(x)dx \qquad \text{and} \int_{\infty}^{\infty} f(x)dx = 1$$

- يسمى التابع $f(x)$ بتابع توزع الكثافة الاحتمالي والذي يعبر عن جميع احتمالات الممكنة للمتحول العشوائي المستمر X. [3 إيضاح]



[6 مثال]

4. توليد الأعداد العشوائية Generating Random Numbers

- تستخدم الأعداد العشوائية في كثير من التطبيقات، فمثلاً:
  - يمكن استخدام الأعداد العشوائية في حساب التكاملات المعقدة.
  - يمكن استخدام الأعداد العشوائية في التحقق من كفاءة الخوارزميات.
  - في المحاكاة تستخدم الأعداد العشوائية لمحاكاة الحالات الحقيقية والتأكد من أداء النظام.
- لقد استعرضنا في الفصل السابق مثالاً على إصلاح آلات في المعمل بوجود عامل إصلاح واحد، وذلك بافتراض أن زمن عمل الآلة ثابت وزمن الإصلاح أيضاً ثابت، ولكن في الحياة العملية زمن الإصلاح أو زمن العمل متغير تبعاً لنوع وطبيعة العطل، إذاً من اجل محاكاة المصنع بشكل واقعي يجب على نظام المحاكاة استخدام أعداد عشوائية.
- لاستخدام الأعداد العشوائية في نظام محاكاة بحيث تتبع تابع توزع احتمال مطلوب يجب أولا التعرف على طرق توليد الأعداد العشوائية المنتظمة باستخدام الحاسب.

1.4. الأعداد العشوائية الزائفة Pseudo-Random Numbers

- لا يوجد في الحقيقة عدد عشوائي واحد وإنما سلسلة من الأعداد العشوائية والتي تتبع توزع احتمال ما. في هذا التوزع نحصل على العدد العشوائي بمحض الصدفة مثلاً وهذا العدد ليس له علاقة بالأعداد الأخرى ضمن السلسلة، بحيث أن أي عدد مولد له احتمال ما يقع ضمن المجال المحدد.
- تسمى الأعداد العشوائية التي تتبع تابع توزع منتظم والمحصورة بين ٠ و الـ 1 بالأعداد العشوائية Random Numbers، بينما تسمى الأعداد العشوائية والتي تتبع أي توزع أخر بالأعداد العشوائية المتغيرة Random Variates.
- يمكن اعتماد طريقة قديمة في توليد الأعداد العشوائية، وهي طريقة التربيع لفون نيومن، حيث تعتمد على حساب مربع العدد العشوائي المولد سابقاً و من ثم اختيار الخانات الوسطى من الناتج لتوليد العدد الجديد. [7 مثال] ،[4 إيضاح] .
- إن معظم الحواسب اليوم تستخدم طريق تعتمد على طريقة التطابقية Congruntial Method حيث تظهر الإحصاءات أن الأعداد المولدة نتيجة لهذه البرامج تظهر بأنها عشوائية تقريبا لذلك تدعى بالأعداد العشوائية الزائفة.

2.4. الطرق التطابقيه Congruential methods

- تعتمد الطرق التطابقية على توليد الأعداد بطريقة محددة وثابتة، أي باستخدام معادلات محددة مسبقا لتوليد سلسلة من الأعداد محددة ومعروفة مسبقاً [5 إيضاح] .
- في الحقيقة سلسلة هذه الأعداد غير عشوائية ولكن يمكن اعتبارها عشوائية إذا اجتازت اختبارات خاصة مصممة لفحص الأعداد العشوائية [6 إيضاح] .
- تعتمد هذه الطريقة على العلاقة التكرارية التالية:

$$x_{i+1} = ax_i + c(\text{mod } m)$$

حيث x_i، a، c، m، هي أعداد موجبة. [7 إيضاح] . [8 مثال] .

- نلاحظ من المثال السابق [8 مثال] أن الأعداد المولدة هي شبه عشوائية وليست عشوائية تماما لان الأعداد تتكرر بعد عدد مرات محددة وان دورة التكرار [8 إيضاح] يمكن آن تتغير عن اختيار قيم مختلفة للثوابت. [9 إيضاح] .
- لتبسيط الطريقة السابقة يمكن تعديل العلاقة التكرارية إلى:

$$x_{i+1} = ax_i(\text{mod}\, m)$$

حيث x_i، a، m، هي أعداد موجبة. [10 إيضاح] .

- إذا كان دور التكرار [8 إيضاح] يساوي إلى m فان مولد المتحولات العشوائية له دور يسمى الدور بالدور الكامل Full Period. وكلما كانت قيمة m كبيرة كان الدور كبيراً.
- للحصول على الدور الكامل يجب أن نحقق الشروط التالية:
  - m و c يجب أن لا يكون لهما عامل مشترك.
  - a=1(mod r) حيث r عدد أولي للعدد m أي يقبل القسمة على نفسه أو 1.
  - a=1(mod 4) إذا كانت m من مضاعفات 4.
  - يمكن اختيار أية قيمة الابتدائية x_0 ، حيث لا تؤثر على عمل توليد السلسلة.

[9 مثال]

- ويجب الانتباه إلى تجنب حدوث طفح في سجلات المعالج عندما تستخدم عمليات الضرب عند حساب العدد العشوائي الجديد، حيث يمكن أن يؤدي ذلك إلى توقف البرنامج آو إهمال البيت الأكثر أهمية في أثناء إجراء العمليات الحسابية، مما يؤدي إلى الحصول على نتائج خاطئة تماماً.

3.4. طرق أخرى

بالرغم من أن الطريقة التطابقية هي الأكثر استخداماً لتوليد سلسلة الأعداد العشوائية إلا انه تم تطوير طرق أخرى لتحسين أدائها من حيث زيادة دور التكرار، وتحسين خصائصها من هذه الطرق:

- الطريقة التطابقية العامة General congruential method.
- المولد التركيبي Composite generator.
- مولد توسورث Tausworthe generator.

1.3.4 الطريقة التطابقية العامة General congruential method

- هذه الطريقة تستخدم معادلة التوليد التالية:

$$x_{i+1} = f(x_i, x_{i-1}, x_{i-2}, ...)(\bmod m)$$

حيث $f(x)$ تابع ما يستخدم لتوليد الرقم x_i من مجموعة من الأعداد العشوائية المولدة سابقاً.

- من التوابع الشهيرة التابع التربيعي والذي يأخذ الشكل التالي:

$$x_{i+1} = a_1 x_i^{\;2} + a_2 x_{i-1} + c$$

حيث a_1, a_2, c ثوابت.

- عند جعل القيم a_1=a_2=1 و c=0 فإننا نحصل على طريقة متوسط التربيع [7 مثال]
- الحالة الخاصة الثانية تدعى الطريقة التطابقية التجميعية والتي تعتمد على العلاقة:

$$f(x_i, x_{i-1}, x_{i-2}, ..., x_{i-k}) = a_1 x_i + a_2 x_{i-1} + \cdots + a_k x_{i-k}$$

حيث تعتبر الطريقة التي تعتمد على: $f(x_i, x_{i-1}) = x_i + x_{i-1}$ من أهم حالتها.

2.3.4 المولد التركيبي Composite generator

- تعتمد هذه الطريقة على الجمع بين طريقتين [9 إيضاح] وذلك لتحسين أداء سلسلة الأعداد العشوائية الناتجة.
- تعتمد هذه الطريقة مثلاً على استخدام مولدين تربيعين. يستخدم الأول لتوليد مجموعة من k عدد عشوائي، تستخدم هذه الأعداد باستخدام المعادلة الثانية لتوليد r عدد عشوائي، هذه الأعداد تستخدم بدورها لتوليد k رقم جديد من الأعداد العشوائية باستخدام المعادلة الأولى، وهكذا تكرر العملية.
- يمكن التحقق من أن هذه الطريقة يمكن أن تعطي نتائج مرضية حتى ولو أن المولدين المستخدمين غير جيدين.

3.3.4 مولد توسورث Tausworthe generator

- تعتمد هذه الطريقة على الطريقة التطابقية التجميعية عندما يكون $m=2$.

$$x_{i+1} = f(x_i, x_{i-1}, x_{i-2}, ...)(\text{mod } 2)$$

حيث

$$f(x_i, x_{i-1}, x_{i-2}, ..., x_{i-k}) = a_1 x_i + a_2 x_{i-1} + \cdots + a_k x_{i-k}$$

أي:

$$x_{i+1} = (a_1 x_i + a_2 x_{i-1} + \cdots + a_k x_{i-k}) \text{ mod } 2$$

حيث x_i تأخذ احد القيمتين 0 أو 1. [11 إيضاح]

- يمكن تبسيط الطريقة باستخدام العلاقة المنطقية EX-OR [12 إيضاح] حيث تستخدم مجموعة البيتات المولدة لتكوين سلسلة من n-bits تشكل الرقم العشوائي المحصور بين القيمة 0 و $2n$-1.

5. طرق اختبار الأعداد العشوائية الزائفة

- إن معظم الطرق المستخدمة لتوليد الأعداد العشوائية الزائفة Pseudo-random numbers هي طرق محددة أي أن الأعداد التي تولد عنها ليست عشوائية. ولكن يمكن اعتبار هذه الأعداد عشوائية طالما أنها اجتازت عدد من الاختبارات الإحصائية.
- يمكن اعتبار سلسلة أعداد بأنها عشوائية إذا كانت تحقق الشرطين التاليين:
  - إذا كانت موزعة بشكل منتظم Uniformly distributed.
  - وكانت مستقلة عن بعضها البعض.
- سوف نشرح طريقتين أساسيتين لاختبار الأعداد العشوائية:
  - اختبارات التشغيل Runs test.
  - اختبار Chi-Square.

1.5 اختبارات التشغيل Runs Test

- يستخدم هذا الاختبار لمعرفة ما إذا كانت سلسلة الأعداد العشوائية الزائفة مستقلة independent أم لا
- تعتمد هذه الطريقة على الخطوات التالية:

1 نأخذ سلسلة من الأعداد العشوائية بين ٠ والـ 1

مثال السلسلة: 0.8, 0.7, 0.75, 0.55, 0.6, 0.7, 0.3, 0.4, 0.5

2 سوف نقسم هذه السلسلة إلى مجموعات من السلاسل صغيرة run up يمكن أن يكون طول الواحدة منها عدد واحد، وبحيث تكون الأعداد العشوائية فيها تتزايد بشكل تصاعدي مثال:

- 0.8 السلسلة الجزئية الأولى بطول 1.

0.7, 0.75 السلسلة الجزئية الثانية بطول 2.

0.55, 0.6, 0.7 السلسلة الجزئية الثالثة بطول 3.

0.3, 0.4, 0.5 السلسلة الجزئية الرابعة بطول 3.

3 ترمز السلاسل الجزئية run up المتساوية الطول بالرمز r_i حيث i طول السلسلة وبحيث تجمع جميع السلاسل $i\geq 6$ مع بعضها البعض.

مثال: $r_1=1,\ r_2=1,\ r_3=2$.

4 تستخدم رموز السلاسل الجزئية r_i بعد ذلك لحساب العلاقة:

$$R=\frac{1}{n}\sum_{1\le i,j\le 6}(r_i-nb_i)(r_j-nb_j)a_{ij},\qquad 1\le i\le 6, 1\le j\le 6$$

حيث n عدد العينات المستخدمة و b_i مجموعة من الثوابت تعطى بالمصفوفة التالية:

$$(b_1,b_2,\quad ,b_6)=(\frac{1}{6},\frac{5}{24},\frac{11}{120},\frac{19}{720},\frac{29}{5040},\frac{1}{840})$$

و a_{ij} ثوابت تعطى بالمصفوفة الثنائية التالية:

$$\begin{bmatrix} 4529.4 & 9044.9 & 13568 & 18091 & 22615 & 27892 \\ 9044.9 & 18097 & 27139 & 36187 & 45234 & 55789 \\ 13568 & 27139 & 40721 & 54281 & 67852 & 83685 \\ 18091 & 36187 & 54281 & 72414 & 90470 & 111580 \\ 22615 & 45234 & 67852 & 90470 & 113262 & 139476 \\ 27892 & 55789 & 83685 & 111580 & 139476 & 172860 \end{bmatrix}$$

إذا عند قيمة لـ $R\geq 4000$ فإن تابع التوزيع الاحتمالي للعدد العشوائي سيكون بشكل Chi-square بدرجة حرية تساوي 6.

2.5 اختبار Chi-Square

- الخطوة التالية لاختبار استقلالية المتحولات المنطقية، هي باختبار توزع المنتظم للمتحولات المنطقية العشوائية.
- يتم اختبار chi-square بإتباع الخطوات التالية:

1. ليكن لدينا سلسلة الأعداد العشوائية المحصورة بين 0 و 1.
2. يتم تقسيم هذه السلسلة إلى k سلسله جزئية بطول متساوي، حيث $k>100$.
3. عدد الأرقام العشوائية ضمن السلسلة الجزئية ذات الترتيب i يساوي f_i بحيث يجب أن يكون $f_i>5$. [13 إيضاح]
4. إذا كانت الأرقام العشوائية ضمن السلسلة الجزئية موزعة بشكل منتظم تماماً، يتم حساب متوسط الأرقام العشوائية ضمن السلسلة الجزئية، n/k حيث n عدد العينات.

[14 إيضاح]

5. يتم بعد ذلك حساب العلاقة التالية:

$$\chi^2 = \frac{k}{n}\sum_{i=1}^{k}(f_i - \frac{n}{k})^2$$

والتي لها k-1 درجة حرية.

6. يتم مقارنة النتيجة مع قيمة محددة مسبقاً ضمن جداول محددة فإذا كانت القيمة χ^2 اكبر من القيمة المفترضة في الجداول ترفض النتيجة ويعتبر التوزيع غير منتظم.

[15 إيضاح]

ملحقات

[1 مثال]

عند رمي قطعة نقدية فإننا نحصل على الفضاء S على احد النتيجتين (H الصورة) أو (T النقش):

$$S = \{\mathrm{H}, \mathrm{T}\}$$

حيث تمثل { } بمجموعة تحتوي على T أو H.

[2 مثال]

عند رمي النرد فإننا نحصل على الفضاء S على احد النتائج التالية:

$$S = \{1,2,3,4,5,6\}$$

[3 مثال]

عند رمي نردين في أن واحد فإننا نحصل على الفضاء التالي:

$$S = \{(1,1),(1,2),(1,3),....,(6,6)\}$$

حيث يمثل الرقم الأول i في المجموعة (i, j) بالرقم الناتج من النرد الأول و j بالرقم الناتج من النرد الثاني. ويمثل المتحول العشوائي X هو ناتج جمع قمتي النردين.

[1 إيضاح]

سوف نرمز للمتحول العشوائي بالحرف اللاتيني الكبير X،Y،Z بينما لقيم المتحول العشوائي بالحرف الصغير x,y,z.

[2 إيضاح]

يتميز تابع توزع الاحتمالي بالخصائص التالية:

- $0 \le F(x) \le 1$ من اجل كل قيم x.
- إن التابع $F(x)$ غير متناقص أي إذا كان $x_1 < x_2$ فان $F(x_1) \le F(x_2)$.
- دوماً $\lim_{x\to\infty} F(x) = 1$ و $\lim_{x\to-\infty} F(x) = 0$.

سوف نرمز للمتحول العشوائي بالحرف اللاتيني الكبير X،Y،Z بينما لقيم المتحول العشوائي بالحرف الصغير x,y,z.

[3 مثال]

نظام الطائرة يعتبر نظام مستمر حيث تتغير معاملات النظام (كتغير علو الطائرة) بشكل مستمر مع الزمن.

[4 مثال]

الأعداد الحقيقية المحصورة بين ٠ و 1.

[5 مثال]

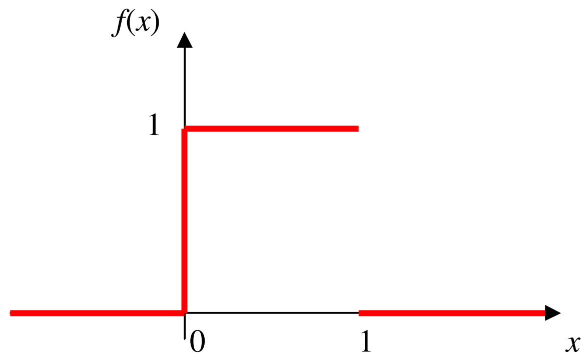
الأعداد الحقيقية الموجبة.

[3 إيضاح]

يعبر تابع الكثافة الاحتمالي للمتحول العشوائي X بأنه عدد مرات تكرار العدد العشوائي x_i.

[6 مثال]

المتحول العشوائي المنتظم بين المجال [0, 1] له تابع كثافة الاحتمالي التالي:



[7 مثال]

لنفترض أن العدد العشوائي القديم هو 5772156649 إذا بحساب المربع نجد النتيجة 09201 7923805949 33317 أي أن الرقم العشوائي الجديد هو: 7923805949.

[4 إيضاح]

من المهم معرفة ما إذا كانت هذه الطريقة تولد سلسلة من الأعداد العشوائية صحيحة بحيث لا يوجد ترابط بين هذه الأعداد، في الحقيقة لا يمكن ملاحظة أن هذه الطريقة لا تولد سلسلة من الأعداد العشوائية.

[5 إيضاح]

أي انك تستطيع معرفة قيمة العدد العشوائي مثلا الخامس قبل توليد السلسة أصلا.

[6 إيضاح]

يتم اختبار انتظام سلسلة الأعداد وعدم ارتباط هذه الأعداد مع بعضها البعض.

[7 إيضاح]

يتم حساب الرقم العشوائي الجديد x_{i+1} من العدد القديم x_i وذلك بضربه أولاً بالثابت a وجمع الناتج مع الثابت c ومن ثم تقسيم الناتج على m واختيار باقي القسمة فقط.

[8 مثال]

مثلاً $x_0=a=c=7$ و $m=10$ إذاً نحصل على:

x_1=mod((7*7+7)/10)=mod(56/10)=mod(5.6)=6,

x_2=mod((6*7+7)/10)=mod(49/10)=mod(4.9)=9,

x_3=mod((9*7+7)/10)=mod(70/10)=mod(7.0)=0,

x_4=mod((0*7+7)/10)=mod(7/10)=mod(0.7)=7,

x_5=mod((7*7+7)/10)=mod(56/10)=mod(5.6)=6,

x_6=mod((6*7+7)/10)=mod(49/10)=mod(4.9)=9,

…

[8 إيضاح]

ويعرف دور التكرار بعدد الأرقام العشوائية (المسافة) التي تولدها الخوارزمية (المعادلة) قبل أن تتكرر مره أخرى.

[9 إيضاح]

قم باختيار قيم أخرى للثوابت a، c، m واحسب السلسلة وحاول آن تجد التكرار.

[10 إيضاح]

نلاحظ أن سلسلة الأعداد المولدة محصورة بين ٠ و m-1 ويمكن الحصول بسهولة على سلسلة الأعداد العشوائية محصورة بين 0 و 1 بتقسيم x/m.

[9 مثال]

إن القيم المثالية المستخدمة في توليد سلسلة الأعداد العشوائية هي:

$$a=314, 159, 269, c=453, 806, 245, m=231$$

[11 إيضاح]

يمكن نلاحظ أن هذه الطريقة تولد سلسلة من البيت $\{b\}$، وذلك عند افتراض أن a_i ذات قيم ثنائية أيضاً

[12 إيضاح]

إن علاقة EX-OR تعطى وفق جدول الحقيقة التالي:

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

[13 إيضاح]

تسمى f بالمراقب observed.

[14 إيضاح]

تعتمد فكرة اختبار chi-square على حساب تغيرات الفرق بين القيمة الملاحظة من القياس الحقيقي (الأعداد المولدة) وبين القيمة النظرية والمفترض الحصول عليها.

[15 إيضاح]

يمكن للطالب تحميل الملف المرفق chi-square.pdf الذي يحتوي على جداول χ^2.

التمارين

5- أستخدم الطريقة التطابقية لتوليد سلسلة من الأعداد العشوائية يدوياً عندما تكون m=10 و c=7 و a=1 ومن ثم احسب دور التكرار.

6- أستخدم الطريقة التطابقية لتوليد سلسلة من الأعداد العشوائية يدوياً عندما تكون m=7 و c=5 و a=3 ومن ثم احسب دور التكرار.

7- أستخدم الطريقة التطابقية لتوليد سلسلة من الأعداد العشوائية يدوياً عندما تكون m=36 و c=5 و a=3 ومن ثم احسب دور التكرار، هل حققت الدور الكامل.

8- أستخدم الطريقة التطابقية العامة لتوليد سلسلة من الأعداد العشوائية يدوياً عندما تكون a_3=7 و a_2=5 و a_1=3 و c=4 ، وابتدأ من القيمة الابتدائية الواحد. من ثم احسب دور التكرار.

9- أستخدم الطريقة التطابقية العامة التجميعية لتوليد سلسلة من الأعداد العشوائية يدوياً عندما تكون القيمة الابتدائية الواحد.

10- استخدم طريقة التوسورث لتوليد سلسلة من الأعداد العشوائية يدوياً عندما تكون a_3=7 و a_2=5 و a_1=3 ، من ثم احسب دور التكرار.

مسائل

4- أكتب برنامجا يقوم بإجراء بتوليد الأعداد العشوائية وفق نفس الشروط في التمرين الأول السابق، ثم ارسم الـ Histogram باستخدام برنامج Excel ثم تأكد من النتائج باستخدام Run test و Chi-test.

5- أكتب برنامجا يقوم بإجراء بتوليد الأعداد العشوائية وفق نفس الشروط في التمرين الثاني السابق، ثم ارسم الـ Histogram باستخدام برنامج Excel ثم تأكد من النتائج باستخدام Run test و Chi-test.

6- أكتب برنامجا يقوم بإجراء بتوليد الأعداد العشوائية وفق نفس الشروط في التمرين الثالث السابق، ثم ارسم الـ Histogram باستخدام برنامج Excel ثم تأكد من النتائج باستخدام Run test و Chi-test.

7- أكتب برنامجا يقوم بإجراء بتوليد الأعداد العشوائية وفق نفس الشروط في التمرين الرابع السابق، ثم ارسم الـ Histogram باستخدام برنامج Excel ثم تأكد من النتائج باستخدام Run test و Chi-test.

8- أكتب برنامجا يقوم بإجراء بتوليد الأعداد العشوائية وفق نفس الشروط في التمرين الخامس السابق، ثم ارسم الـ Histogram باستخدام برنامج Excel ثم تأكد من النتائج باستخدام Run test و Chi-test.

9- أكتب برنامجا يقوم بإجراء بتوليد الأعداد العشوائية وفق نفس الشروط في التمرين السادس السابق، ثم ارسم الـ Histogram باستخدام برنامج Excel ثم تأكد من النتائج باستخدام Run test و Chi-test.

توليد الأعداد العشوائية

Generating Stochastic Varieties

1. مقدمة

- لقد استعرضنا في الفصل السابق طرق توليد الأعداد العشوائية المحددة (الزائفة) والموزعة بشكل منتظم random numbers.
- في هذا الفصل سوف ندرس طرق توليد الأعداد العشوائية الغير محددة والتي تسمى random variates أو stochastic variates.
- حيث تلعب الأعداد العشوائية المحددة دوراً مهماً في توليد الأعداد العشوائية الغير محددة.

2. طريقة التحويل العكسي The inverse transform method

- تستخدم هذه الطريقة فقط عندما يكون حساب معكوس تابع توزيع الكثافة الاحتمال ممكن من الناحية العملية.
- لنفترض أننا نرغب بتوليد أعداد عشوائية غير محددة لها تابع كثافة احتمالي Probability Density Function (pdf) مساوياً $f(x)$. ولنفترض أن تابع تراكم الكثافة الاحتمالي Cumulative density function يساوي $F(x)$ ومحصور ضمن المجال 0 و 1.
- نقوم أولا بتوليد رقم عشوائي r بحيث:

$$F(x) = r$$

- ثم نقوم بالحصول على العدد العشوائي المطلوب x بحساب مقلوب F أي:

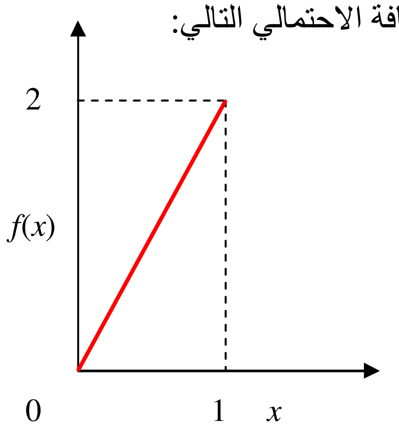
$$x = F^{-1}(r)$$

- حيث $F^{-1}(r)$ مقلوب التابع F.

مثال:

- لنفترض أننا نريد توليد أرقام عشوائية غير محددة لها تابع الكثافة الاحتمالي التالي:

$$f(x) = 2x, \qquad 0 \le x \le 1$$



- نقوم أولا بحساب تابع الكثافة التراكمي $F(x)$ باستخدام العلاقة:

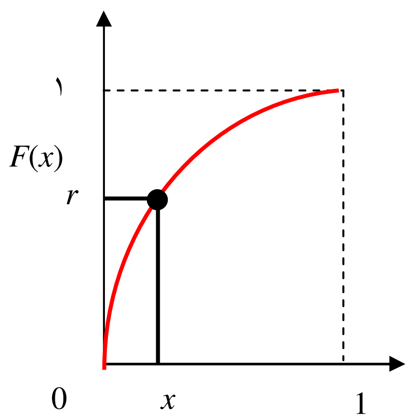
$$F(x) = \int_a^b f(t)dt$$

أي:

$$F(x) = \int_0^x 2tdt = x^2 \qquad 0 \le x \le 1$$

- أي أن الأعداد العشوائية المطلوبة تساوي:

$$r = x^2 \qquad \Rightarrow \qquad x = \sqrt{r}$$



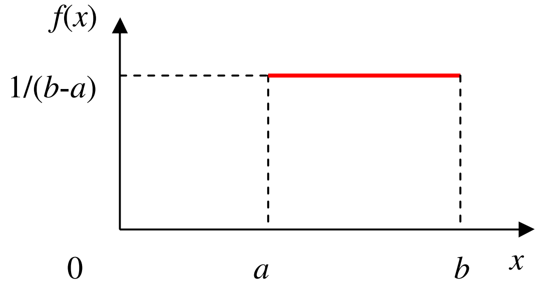
3. توليد الأعداد العشوائية من توابع الاحتمالية المستمرة

- في هذه الفقرة سوف نستخدم طريقة التحويل العكسي لتوليد الأعداد العشوائية الغير محددة من مجموعة من توابع توزيع كثافة احتمالية مختلفة وهي:
  - من تابع توزع منتظم uniform distribution.
  - تابع توزع أسي exponential distribution.
  - تابع توزع إيرلانغ Erlang distribution.
  - تابع توزع طبيعي Normal distribution.

1.3 توليد الأعداد العشوائية من تابع توزع منتظم

- إن تابع الكثافة الاحتمالي للتوزع المنتظم يعطى بالعلاقة:

$$f(x)=\begin{cases}\dfrac{1}{b-a} & a\le x\le b\\ 0 & otherwise\end{cases},$$



- نقوم بحساب تابع الكثافة التراكمي $F(x)$ باستخدام العلاقة:

$$F(x)=\int_a^x \frac{1}{b-a}dt=\frac{1}{b-a}\int_a^x dt=\frac{x-a}{b-a}$$

- إن القيمة المتوقعة expectation (المتوسط المتوقع) والانحراف variance يعطيان بالعلاقتين:

$$E(x)=\int_a^b f(x)xdx=\frac{1}{b-a}\int_a^b xdx=\frac{b+a}{2}$$

$$Var(x)=\int_a^b (x-E(x))^2 F(x)dx=\frac{(b-a)^2}{12}$$

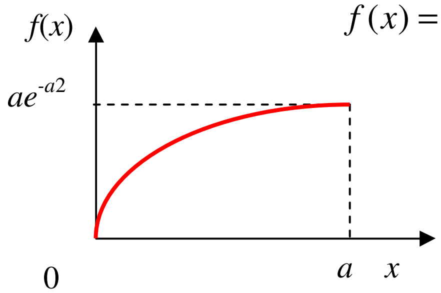
- بحساب معكوس التابع $F(x)$ نحصل على:

$$r=F(x)=\frac{x-a}{b-a}\quad\Rightarrow\quad x=a+(b-a)r$$

2.3 توليد الأعداد العشوائية من تابع توزع أسي

- إن تابع الكثافة الاحتمالي للتوزع الأسي يعطى بالعلاقة:

$$f(x) = ae^{-ax}, \qquad a > 0, x \geq 0$$



- نقوم بحساب تابع الكثافة التراكمي $F(x)$ باستخدام العلاقة:

$$F(x) = \int_0^x f(t)dt = \int_0^x ae^{-at}tdt = 1 - e^{-ax}$$

- إن القيمة المتوقعة expectation والانحراف variance يعطيان بالعلاقتين:

$$E(x) = \int_0^\infty aet^{-at}tdtdt = \frac{1}{a}$$

$$Var(x) = \int_0^\infty (t - E(x))^2 e^{-at} tdtdt = \frac{1}{a^2}$$

- بحساب معكوس التابع $F(x)$ نحصل على:

$$r = F(x) = 1 - e^{-ax} \qquad \Rightarrow \quad 1 - r = e^{-ax}$$

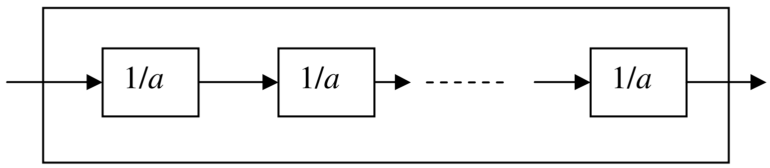
$$x = -\frac{1}{a}\log(1-r) = -E(x)\log(1-r)$$

- باعتبار أن $F(x)$ موزعة بشكل منتظم بين 0 و 1 نجد:

$$r = e^{-ax} \qquad \Rightarrow \quad x = -\frac{1}{a}\log r$$

3.3 توليد الأعداد العشوائية من تابع إيرلانغ Erlang

- يستخدم في معظم نظم المحاكاة الأعداد العشوائية الغير محددة والتي تتبع تابع توزع الاحتمالي الأسي ولكنها لا تغطي الحالة الواقعية لتغير متحولات دخل نظام المحاكاة.
- بينما تتبع الحالات الواقعية تغيرات متحولات بتوزع احتمالي يساوي إلى مجموع توابع أسية تتابع تسلسلياً.
- عندما يكون متوسط mean هذه التوابع الأسيه واحد تقريباً، فان تابع التوزيع الكلي الناتج يساوي إلى تابع توزيع إيرلانغ Erlang.



- إن تابع الكثافة الاحتمالي للتوزع إيرلانغ هو عبارة عن جداء طي لـ k تابع توزع أسي له متوسط mean يساوي إلى $1/a$.
- إذا القيمة المتوقعة expectation والانحراف variance عبارة عن تكرار القيمة المتوقعة والانحراف المعياري بعدد k مرة أي:

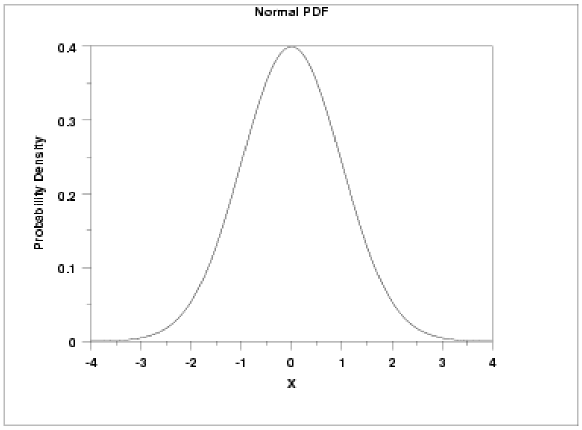
$$E(x) = \frac{k}{a}$$

$$Var(x) = \frac{k}{a^2}$$

- يتم حساب العدد العشوائي المطلوب من تابع توزيع الأسي وذلك عن طريق حساب مجموع لـ k متحول عشوائي ذو تابع توزيع أسي أي:

$$x = \sum_{i=1}^{k} x_i = -\frac{1}{a}\sum_{i=1}^{k}\log r_i = -\frac{1}{a}\left(\log \sum_{i=1}^{k} r_i\right)$$

4.3 توليد الأعداد العشوائية من تابع توزع الطبيعي



- إن تابع الكثافة الاحتمالي للتوزع الطبيعي يعطى بالعلاقة:

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}, \qquad -\infty<x<+\infty$$

حيث أن القيمة المتوقعة تساوي μ والانحراف يساوي σ^2.

[1 إيضاح] .

- لتوليد أعداد عشوائية ذات تابع كثافة احتمالي طبيعي يجب استخدام نظرية المركز المحدد [2 إيضاح]
- إذا بتوليد k عدد عشوائي منتظم r_1، r_2، .... r_k محصورة بين 0 و 1، نحصل على:

$$E(r_i)=\frac{a+b}{2}=\frac{1}{2}$$

$$Var(r_i)=\frac{(b-a)^2}{12}=\frac{1}{12}$$

- بتطبيق نظرية المركز المحدد [2 إيضاح] نحصل على:

$$\sum r_i \approx N(\frac{k}{2},\frac{k}{\sqrt{12}})$$

حيث N ترمز للتوزع المنتظم، وبالتالي تصبح العلاقة:

$$\frac{\sum r_i - k/2}{k/\sqrt{12}} \approx N(0,1)$$

- لنفترض أن x توزع طبيعي فان:

$$\frac{x-\mu}{\sigma} \approx N(0,1) \quad \Rightarrow \quad \frac{x-\mu}{\sigma} = \frac{\sum r_i - k/2}{k/\sqrt{12}}$$

أي:

$$x = \sigma\sqrt{\frac{12}{k}}\left(\sum r_i - \frac{k}{2}\right) + \mu$$

[3 إيضاح] ، [4 إيضاح] .

4. توليد الأعداد العشوائية من توابع الاحتمالية المتقطعة

- في هذه الفقرة سوف نستخدم طريقة التحويل العكسي لتوليد الأعداد العشوائية الغير محددة من توابع توزيع مختلفة وهي:
  - من تابع توزع الهندسي geometric distribution.
  - من تابع توزع الثنائي binomial distribution .
  - تابع توزع بويسن poisson distribution.

1.4. توليد الأعداد العشوائية من تابع توزع الهندسي

- ليكن لدينا سلسلة من مجموعة من التجارب نتيجتها أما نجاح أو فشل (صح أو خطأ).
- لنرمز لاحتمال النجاح بـ p و لاحتمال الخطأ بـ q على الترتيب و بحيث $p+q=1$. إن المتحول العشوائي لتتابع الأخطاء يتبعها نجاح واحد يتبع تابع توزع احتمالي هندسي.
- إن تابع الكثافة الاحتمالي للتوزع الهندسي يعطى بالعلاقة:

$$p(n) = pq^n, \qquad n = 0,1,2,3,....$$

اي ان تابع الكثافة التراكمي $F(n)$ باستخدام العلاقة:

$$F(n) = \sum_{s=0}^{n} pq^s, \qquad n = 0,1,2,3,...$$

- إن القيمة المتوقعة expectation والانحراف variance يعطيان بالعلاقتين:

$$E(x) = \frac{p}{q}$$

$$Var(x) = \frac{p}{q^2}$$

- بحساب معكوس التابع $F(n)$ نحصل على:

$$F(n) = \sum_{s=0}^{n} pq^s = p\sum_{s=0}^{n} q^s = p\frac{1-q^{n+1}}{1-q}$$

وبما أن $p=1-q$ أي:

$$F(n) = 1-q^{n+1} \Rightarrow 1-F(n) = q^{n+1}$$

أي:

$$r = q^{n+1} \Rightarrow \log r = (n+1)\log q \Rightarrow n = \frac{\log r}{\log q} - 1$$

- وباعتبار أن $(1-F(n))/q=q^n$، و $(1-F(n))/q$ يتغير بين 0 و 1 نحصل على:

$$r = q^n \Rightarrow n = \frac{\log r}{\log q}$$

2.4. توليد الأعداد العشوائية من تابع توزع الثنائي

- ليكن لدينا سلسلة من مجموعة من التجارب نتيجتها إما نجاح أو فشل (صح أو خطأ).
- لنرمز لاحتمال النجاح بـ p و ولاحتمال الخطأ بـ q على الترتيب و بحيث $p+q=1$. ولنفترض أن المتحول العشوائي X هو متحول لتتابع ظهور حالات النجاح، إذا فالمتحول العشوائي X يتبع تابع توزع احتمالي الثنائي.
- إن تابع الكثافة الاحتمالي للتوزع الثنائي يعطى بالعلاقة:

$$p(k) = \binom{n}{k} p^{k} q^{n-k}, \qquad k = 0,1,2,3,....$$

- إن القيمة المتوقعة expectation والانحراف variance يعطيان بالعلاقتين:

$$E(x) = np$$

$$Var(x) = npq$$

- لتوليد المتحول العشوائي X المطلوب يتم عن توليد n رقم عشوائي بجعل $k_0=0$ ثم يتم زيادة المتحول k_i حسب العلاقة:

$$k_i = \begin{cases} k_{i-1} + 1 & \text{if } r_i < p \\ k_{i-1} & \text{if } r_i > p \end{cases}$$

إن المتحول الناتج k_n هو المتحول العشوائي المطلوب.

3.4. توليد الأعداد العشوائية من تابع توزع بويسن

- لنفرض أن λ متوسط عدد مرات الظهور خلال فترة محددة من الزمن، إذا عدد مرات الظهور x خلال فترة واحدة الزمن يتبع تابع الكثافة الاحتمالي لتوزع بويسن التالي:

$$p(n) = e^{\ \lambda}(\lambda^{n} / n!), \qquad n = 0,1,2,3,....$$

[5 إيضاح]

- إن الطريقة المتبعة لتوليد عدد عشوائي بتابع توزع كثافة احتمالي بتوزع بويسن تتم بتوليد متحولات عشوائية بتابع توزع أسي بقيمة متوقعة $1/\lambda$ ضمن فترة زمنية محددة ثم جمع المتحولات ضمن واحدة الزمن:

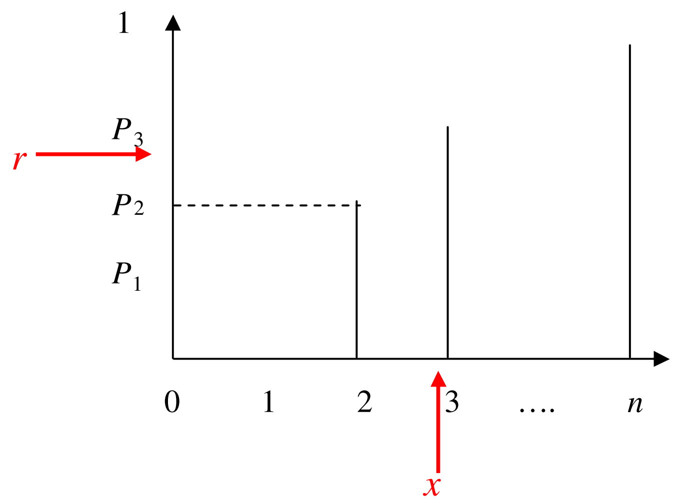
$$\sum_{i=1}^{n} t_i < 1 < \sum_{i=1}^{n+1} t_i$$

5. توليد الأعداد العشوائية من توابعِ الاحتمالية التجريبية

- في كثير من الأحيان الأعداد العشوائية المولدة ليس لها تابع توزع احتمالي معروف، في هذه الحالة يتطلب من نظام المحاكاة توليد أعداد عشوائية تتبع توابع احتمالية ذات توزيع مطابق للقيم للتجريبية.
- في هذا القسم سوف نستعرض طريقتين لتوليد الأعداد العشوائية ذات تابع توزيع تجريبي وذلك من توابع احتمالية مستمرة أو متقطعة.

1.5 تولِيد الأعداد العشوائية التجريبية من توابعِ الاحتمالية المتقطعة

- ليكن لدينا المتحول العشوائي X، و $p(X=i)=p$ حيث p_i تحسب من معلومات تجريبية وجد انه يمكن توليد العدد العشوائي التجريبي بإتباع القاعدة التالية:
- ليكن لدينا r عدد عشوائي ما يقع بين p_3 و p_2 إذا فان الرقم العشوائي x يساوي إلى 3. [6 إيضاح]



مثال:

- ليكن لدينا المتحول العشوائي X والتي تمثل أعداد الصحف التي توزع يومياً، حيث وجدد أن توزع توزيع الصحف على النحو التالي:

	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس
X	1	2	3	4	5
$f(x)$	0.20	0.20	0.30	0.15	0.15

- إذا فان تابع التوزيع التراكمي يساوي [7 إيضاح] :

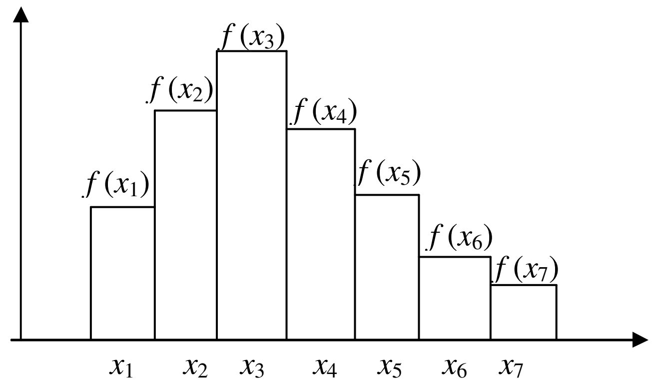
X	1	2	3	4	5
$f(x)$	0.20	0.40	0.70	0.85	1.0

- إن العدد العشوائي التجريبي r والذي يتبع التوزيع السابق يساوي إلى نهاية المجال المحدد تجريبياً كما يلي:

  - If $0.85<r\leq1.00$ then $x=5$
  - If $0.70<r\leq0.85$ then $x=4$
  - If $0.40<r\leq0.70$ then $x=3$
  - If $0.20<r\leq0.40$ then $x=2$
  - Otherwise $x=1$

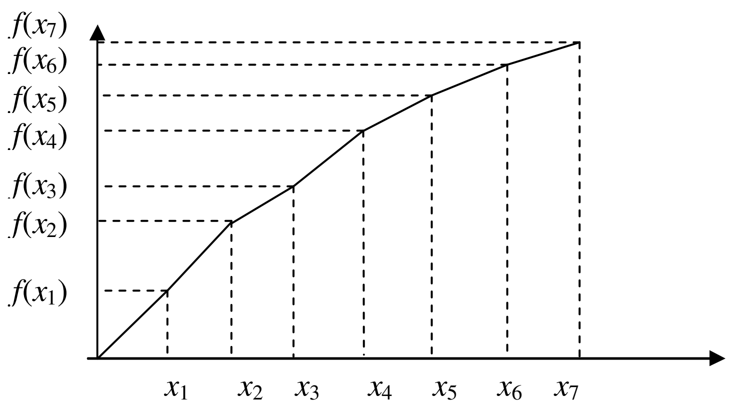
2.5 توليد الأعداد العشوائية التجريبية من توابع الاحتمالية المستمرة

- ليكن لدينا المتحول العشوائي X، والذي يمكن التعبير عنه بالهيستوغرام Histogram التالي:



حيث تمثل النقطة x_i منتصف المجال i، ويمثل $f(x_i)$ مطال المجال i.

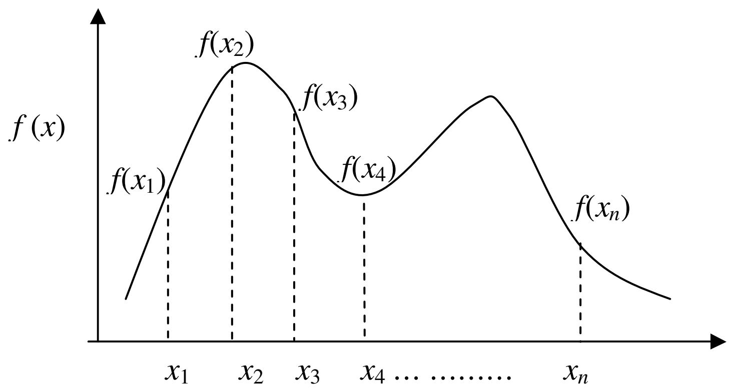
- باستخدام القيم السابقة يمكن بسهولة الحصول على تابع التوزع التراكمي [٧ إيضاح] حيث افترضنا إن تابع التوزيع التراكمي تصاعدياً. إن تابع التوزع التراكمي السابق يبين وفق الشكل التالي:



- ليكن لدينا الآن العدد العشوائي r، ولتفترض أن $f(x_i\text{-}1)<r<f(x_i)$ وباستخدام الإقحام الخطي linear interpolation، فان المتحول العشوائي x نحصل عليه باستخدام العلاقة:

$$x = x_{i-1} + (x_i - x_{i-1})\frac{r - f(x_{i-1})}{f(x_i) - f(x_{i-1})}$$

حيث x_i القيمة العظمى من المجال.



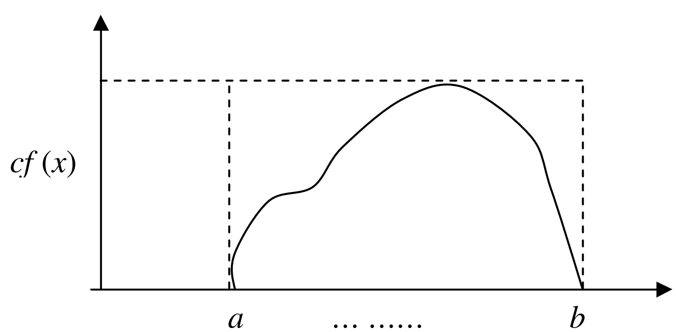
6 طريقة الرفض Rejection methods

٥. يمكن استخدام طريقة الرفض في توليد متحول عشوائي x بتابع توزع مطلوب $f(x)$ وذلك بإتباع الخطوات التالية:

- ليكن المتحول x محصور ضمن المجال $a \leq x \leq b$.
- يتم تحديد قيمة $f(x)$ بحيث أن معامل ضرب c يجعل قيمة التابع:

$$cf(x) \leq 1, \quad .a \leq x \leq b$$

- لنحدد المتحول x كتابع خطي لـ r بحيث يكون $x=a+(b-a)r$، حيث r عدد عشوائي.
- نقوم بتوليد أزواج من الأعداد العشوائية $(r_2، r_1)$.
- بعد ذلك نقوم باختبار أزواج الأعداد العشوائية باستخدام المعادلة $x=a+(b-a)r_1$ ونقوم باختيار الأعداد التي توافق العلاقة $r_2 \leq cf(a+(b-a)r_1)$ والتي تقع تحت المنحني المبين في الشكل.



مثال-1:

استخدم طريقة الرفض لتوليد متحول عشوائي x له تابع توزع كثافة احتمالي يساوي $f(x)=2x,$ $0\leq x\leq 1$

الحل:

١. يتم اختيار قيمة معامل ضرب c بحيث $0\leq x\leq 1$. $cf(x)\leq 1,$ أي $c=0.5$.

٢. نقوم بتوليد العدد العشوائي r_1 ونجعل $x=r_1$.

٣. نقوم بتوليد العدد العشوائي r_2 إذا كان: $r_2\leq cf(r_1)=0.5(2r_1)=r_1$ نقوم بقبول r_2 وإلا نقوم برفضها ونعود للخطوة 2.

مثال-2:

استخدم طريقة الرفض لحساب مساحة دائرة ذات قطر واحدي.

الحل:

1. إن قيمة العددين العشوائيين $r_1^2+r_2^2=1$.
2. نقوم بتوليد العددين العشوائيين r_1 وr_2.
3. إذا كان: $r_2\leq f(r_1)$ حيث $f(r_1)=\sqrt{1-r_1^2}$ إذا فان r_2 تحت (أو على) المنحني وبالتالي يتم قبول الزوجين r_1 و r_2 وإلا نقوم برفضها ونعود للخطوة 2.
4. ويتم حساب المساحة باستخدام المعادلة:

$$area=\frac{\text{total number of acceptable pairs}}{\text{total number of generated pairs}}$$

[**8** إيضاح]

7 طريقة مونتيكارلو Monte Carlo Method

- تعتبر طريقة مونتيكارلو كجزء من الطريقة التجريبية حيث تركز على توليد الأعداد العشوائية التجريبية.
- تعتبر طريقة مونتيكارلو من أكثر الطرق استخداماً عندما تكون النتائج تجريبية أو تستخدم في حال الرغبة في توقع هذه النتائج.
- في الفقرة السابقة وجدنا أن طريقة الرفض قد استخدمت في حساب قيمة تكامل ما عن طريق حساب احتمال النقاط الموجودة تحت منحني التكامل. هذه الطريقة تعتبر احد تطبيقات مونتيكارلو أيضاً.
- ويمكن استخدام مونتيكارلو أيضا في حساب التكامل باستخدام طريقة مختلفة. فلنفترض أننا نريد حساب التكامل التالي:

$$\theta = \int_0^1 f(x)dx$$

- ولنفترض أن ξ_1 ، ξ_2...... ξ_n أعداد عشوائية بين 0 و 1، إذاً التوابع $f(\xi_i)=f_i$ عبارة عن متحولات عشوائية مستقلة بتوقع θ إذاً:

$$\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f_i$$

توقع الكلي للأعداد العشوائية.

- إن توقع الانحراف يساوي إلى:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(f - \bar{f})^2$$

- اذا كان لدينا في الحالة العامة المتحول العشوائي X يأخذ القيم [0, 1] وتابع المتحول العشوائي هو $f(x)$ فان:

$$E(f(x)) = \int_0^1 f(x)g(x)dx$$

- حيث $g(x)$ تابع الكثافة للمتحول X، وبافتراض أن X ذو تابع بتوزع منتظم فان $g(x)=1$ أي نحصل:

$$E(f(x)) = \int_0^1 f(x)dx$$

أي أن:

$$E(f(x)) = f$$

ملحقات

[1 إيضاح]

عندما يكون μ=0 و σ =1 فان تابع التوزيع يسمى تابع كثافة التوزيع الطبيعي القياسي ويعطى بالعلاقة:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}, \qquad -\infty < x < +\infty$$

[2 إيضاح]

تعتمد نظرية المركز المحدد على انه إذا كان لدينا x_1، x_2، ....، x_n متحول عشوائي مستقل وجميعها لها تابع توزيع واحد بتوقع $E(x_i)=\mu$ و انحراف $Var(x)=\sigma^2$ إذا فالمجموع:

$$\sum x_i = x_1 + x_2 + \ldots + x_n$$

هو تقريبا متحول عشوائي بتوزع كثافة احتمالي طبيعي، ويقترب تابع التوزع إلى الطبيعي أكثر كلما زادت n. عندها فان التوقع و الانحراف للمتحول الناتج يساوي إلى:

$$E(\sum x_i) = n\mu$$

$$Var(\sum x_i) = n\sigma^2$$

[3 إيضاح]

هذه المعادلة تستخدم لتوليد الأعداد العشوائية الغير محددة بتابع توزع كثافة احتمال طبيعي.

[4 إيضاح]

إن قيمة k يجب أن تكون كبيرة حتى تكون الدقة اكبر، وقد وجد أن اقل قيمة يمكن استخدامها هي k=12.

[5 إيضاح]

يمكن ملاحظة أن النافذة الزمنية التي تتداخل فيها فترات النجاح تساوي إلى تابع توزع احتمالي بشكل أسي و أن قيمة المتوسط تساوي $1/\lambda$.

[6 إيضاح]

في الحالة العامة إذا كان $p_{i-1}<r<p_i$ فان $x=i$

[7 إيضاح]

تابع التوزع التراكمي يمكن الحصول عليه من العلاقة:

$$F(x_i)=\sum_{1\le k\le n} f(x_k)$$

[8 إيضاح]

إن طريقة الرفض غير دقيقة في توليد العدد العشوائي وخصوصاً عند $c(b-a)$ ذات قيم كبيرة

مسائل

11- قم بكتابة برنامج بلغة عالية المستوى لتوليد الأعداد العشوائية الغير محددة المذكورة في هذا الفصل.

12- قم باختبار الأعداد العشوائية الغير محددة المولدة باستخدام الطرق المذكورة في هذا الفصل.

13- أستخدم البرنامج الذي قمت بتصميمه في الفصل الأول لمحاكاة نظام إصلاح الآلات في مصنع، وقم بتعديل زمن الإصلاح و زمن العمل متحول عشوائي غير محدد بتوزيع احتمالي أسي.ثم نفذ المحاكاة وأطبع النتائج.

تصميم المحاكاة

Simulation Design

1. مقدمة

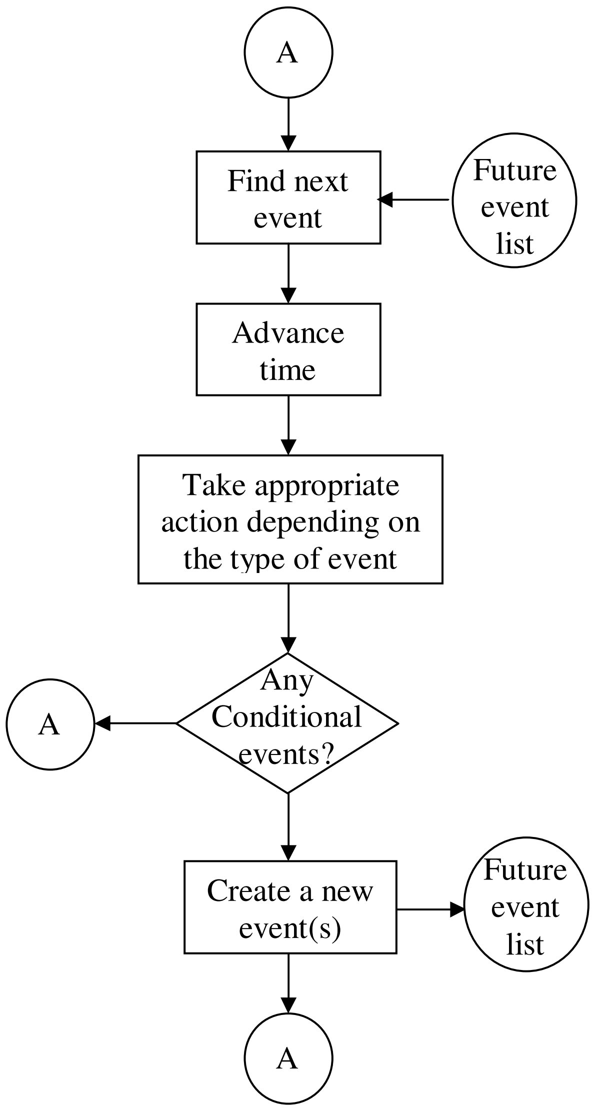
- في هذا الفصل سوف نستعرض ثلاث طرق تستخدم في بناء محاكي
- الطريقة الأولى تسمى تقدم الحدث event-advance، والثانية تسمى تقدم واحدة الزمن unit-time advance، و تعتمدان على تغير حدث المحاكاة ولكنهما تختلفان في طريقة حساب تغيرات الزمن.
- تعتمد الطريقة الثالثة على أسس الأنشطة activities based حيث تعتبر من أشهر طرق المحاكاة المستخدمة.

2. طريقة تقدم الحدث Event-advance design

- تعتمد هذه الطريقة على تحديد ومراقبة الأحداث التي تتغير مع الزمن [1 إيضاح] .
- الفكرة الأساسية من هذه الطريقة تعتمد على تغير حالة النظام status كل مرة يتغير فيها حدث ما في النظام، وبحيث تبقى حالة النظام ثابتة بين حدثين متتابعين. [2 إيضاح]
- لتحقيق ذلك يتم استخدام عداد ساعة clock لكل حدث في النظام، حيث يشير عداد الساعة إلى وقت ظهور الحدث التالي
- ويقوم نظام المحاكي عند الانتهاء من تنفيذ حدث ما (مثلاً t):
  1. بتجميع الأحداث الممكن ظهورها بالمستقبل،
  2. يقوم بترتيب هذه الأحداث تبعاً لزمن الحدوث [3إيضاح]
  3. ثم يبدأ بالحدث الأقل زمنا.
  4. يقوم بتقديم عداد الساعة الرئيسي Master clock إلى زمن يساوي زمن ظهور الحدث الجديد الأقرب (مثلا t_2)
  5. ثم يقوم بتكرار العملية إلى الحدث الأقرب التالي (الزمن t_3) وهكذا .

  [4إيضاح] [1 مثال]

- يمكن لظهور الأحداث الرئيسية primary events [1 مثال] أن يؤدي إلى نشوء أحداث جديدة تسمى أحداث شرطية conditional events [2 مثال] .
- يبين المخطط التدفقي التالي الخطوات الأساسية لطريقة تقدم الحدث:



[5 إيضاح]

3. قائمة الأحداث المستقبلية Future event list

- تعرف قائمة الأحداث المستقبلية Future event list بأنها القائمة التي تحتوي على جدول لجميع الأحداث التي سوف تحدث في المستقبل (أي بعد الزمن *t*).
- أي أن القائمة سوف تحتوي على:
  - الزمن الذي يحدث فيه الحدث Time of occurrence.
  - طبيعة الحدث Type of event [6 إيضاح] .

[3 مثال]

- يجب أن ترتب القائمة الأحداث المستقبلية بحيث تقلل عدد العمليات الحسابية التي يمكن أن يجريها المحاكي للبحث عن الحدث القادم، أي يجب أن تحقق:
  - يجب أن تبين زمن الحدث الجديد (القادم) و طبيعة الحدث.
  - مسح المعلومات الخاصة بالأحداث التي انتهت (نفذت).
  - تتمكن من إضافة حدث جديد إلى القائمة (عند ظهور حدث شرطي مثلاً) أو تعديل في القائمة.
- سوف نبين في الفقرة القادمة طريقتين مستخدمتين في تحقيق قائمة الأحداث المستقبلية:
  - طريقة المصفوفة التسلسلية Sequential array method.
  - طريقة القوائم المترابطة Linked lists method

1.3 طريقة المصفوفة التسلسلية Sequential array method

- تستخدم المصفوفة التسلسلية لتخزين أوقات حدوث الأحداث بشكل تسلسلي في المصفوفة.
- إن أسهل طريقة لتحقيق ذلك هي بتخصيص كل حدث برقم i يعبر عن موقع التخزين في المصفوفة، بينما يخزن في الموقع نفسه زمن حدوث الموقع.
- الشكل التالي يبين مثالاً على طريقة تخزين المعلومات باستخدام المصفوفة، حيث يستخدم الموقع الأول في المصفوفة لتخزين زمن الحدث الأول CL_1 وزمن الحدث الثاني في الموقع الثاني ويساوي CL_2 وهكذا.

CL_1	CL_2	CL_3	. . . . .	CL_n

- إن تحديد الحدث التالي الممكن حدوثه يتم بسهولة عن طريق تحديد موقع المصفوفة التي تحتوي على أقل زمن. يتم ذلك باستخدام خوارزمية بسيطة التالية:

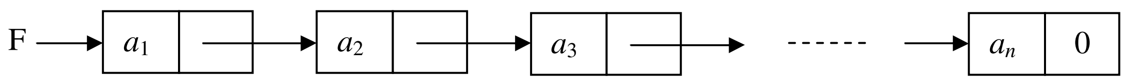
```
num =1
time = A(1)
for i = 1 to n then
    if time > A(i) then
       time = A(i)
       num = i
```

وبالتالي فان المتحول num سوف يحتوي على موقع الحدث التالي والذي يحتوي على اقل زمن.

- المشكلة أن هذه الطريقة لا تحذف الأحداث التي تم تنفيذها، ويتم حل المشكلة بتغيير قيمة الزمن المخزن في المصفوفة للحدث المنفذ بقيمة كبيرة بحيث لا تؤدي إلى اختيار الحدث المنفذ مرة أخرى. ويتم أيضاً إضافة حدث جديد j إلى القائمة عن طريق تغيير الزمن $A(j)$.
- تتميز هذه الطريقة بسهولتها وذلك عند إضافة أو حذف أية حدث.
- إن من أهم مساوئها تكمن بدرجة تعقيدها العالي في البحث عن الحدث التالي، حيث يحتاج ذلك إلى البحث عن أصغر عنصر في المصفوفة، وتكون المشكلة أكبر عندما يكون عدد عناصر المصفوفة كبيراً. لذلك يتم استخدام القوائم المترابطة.

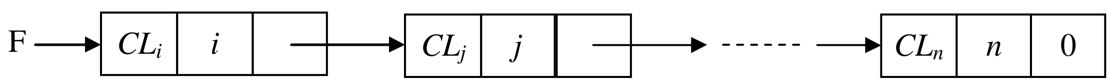
2.3 طريقة القوائم المترابطة Linked lists method

- تعتمد طريقة القوائم المترابطة على تخزين المعلومات عشوائياً ضمن أجزاء مختلفة في الذاكرة وعلى شكل عقد nodes بحيث تحتوي العقدة على جزئين الأول يحتوي على المعلومات المراد تخزينها، ويحتوي الجزء الثاني (المؤشر pointer) على موقع العقدة التالية.



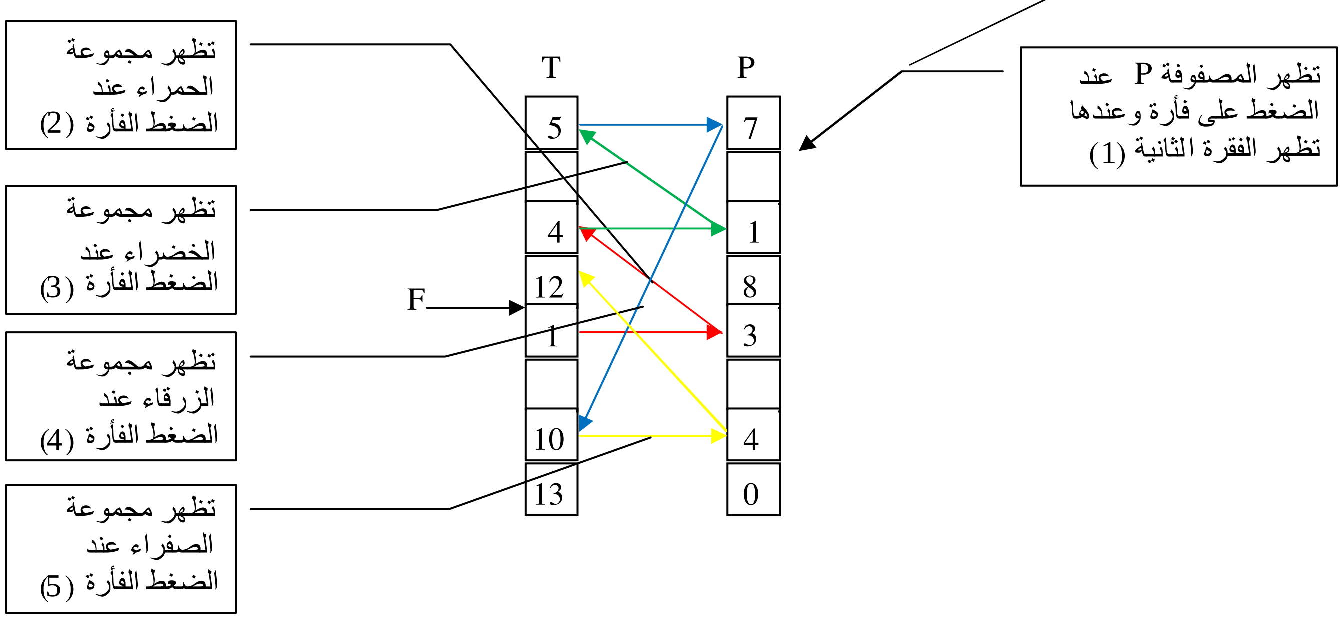
[7 إيضاح]

- يمكن لهذه القائمة أن تستخدم كقائمة الأحداث المستقبلية، حيث يقسم جزء المعلومات إلى قسمين الأول لتخزين الزمن Clock ويستخدم الثاني للحدث.

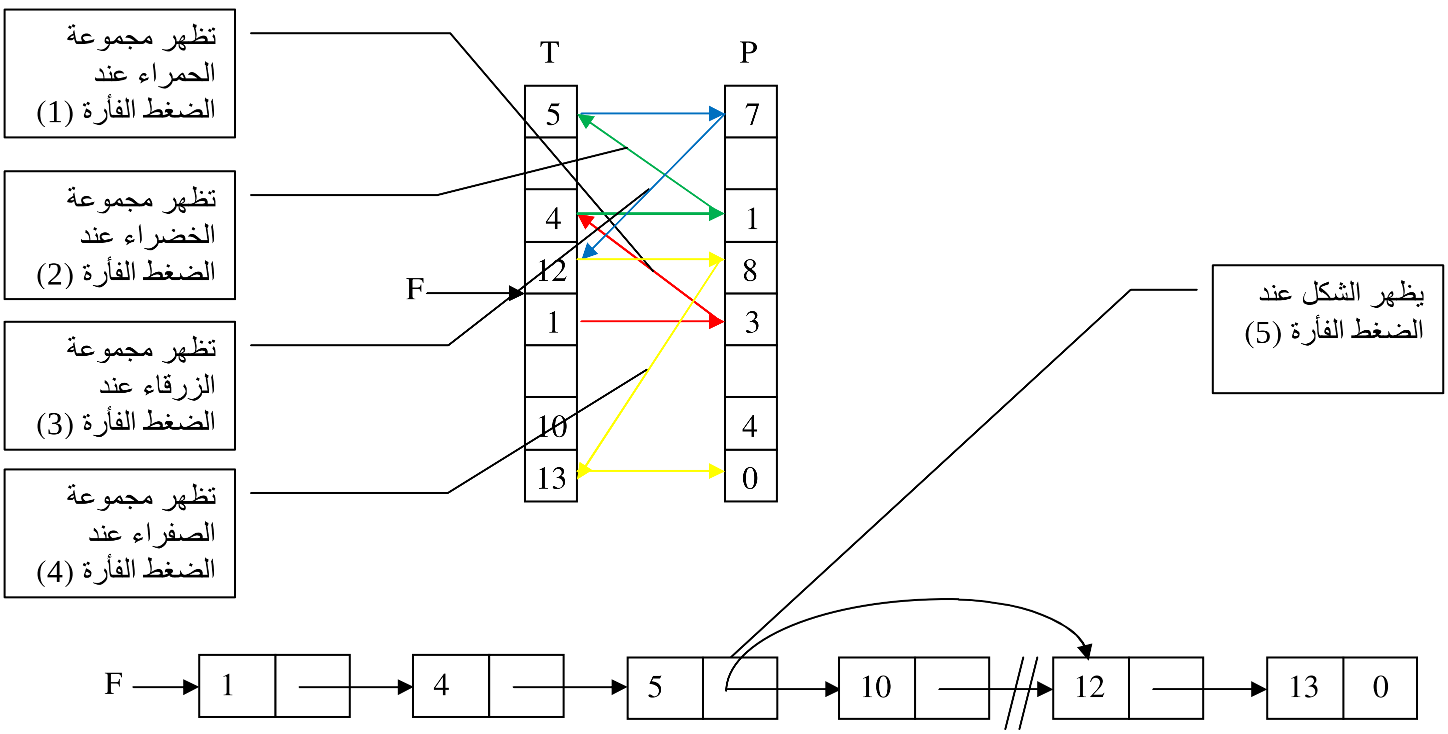


مثال:

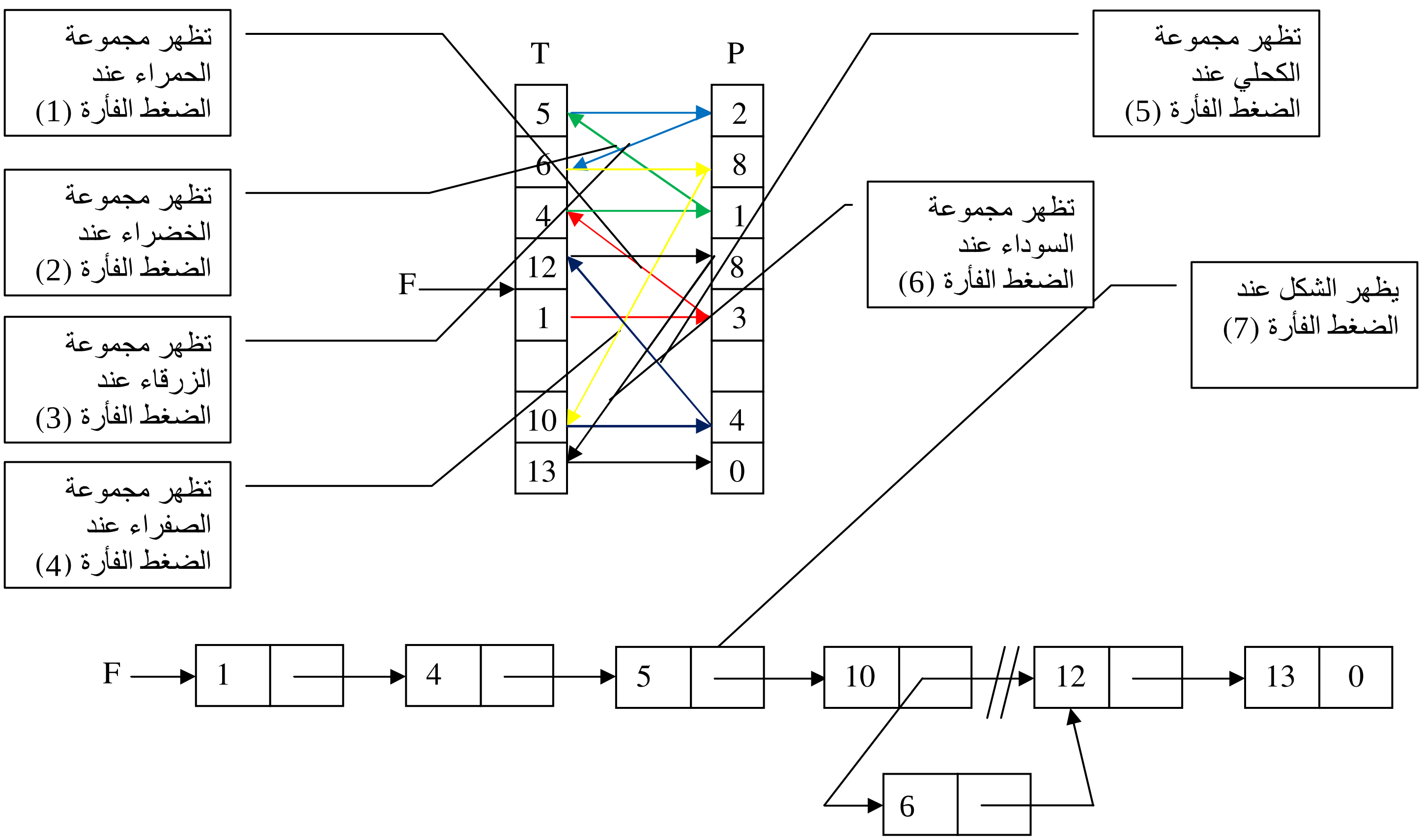
- لنفترض أننا نرغب في ترتيب مجموعة أرقام تصاعدياً والمخزنة بطريقة عشوائية ضمن مصفوفة T.
- لتحقيق ذلك فإننا نحتاج إلى مصفوفة جديدة P لها نفس الطول بحيث أن تسلسل الموقع التالي في T يجب أن يحدد في P وبحيث يتم ترتيب المصفوفة تصاعديا. فمثلاً P(1)=7 أي أن الرقم التالي الأكبر يقع في الموقع السابع في T.



- إن عملية الإضافة أو الحذف من القوائم المترابطة سهل جداً.
- لنفترض أننا نرغب في حذف الرقم 10، نقوم أولا بفحص أول موقع ((T(5), P(5) والذي يشير إلى موقع النقطة التالية ((T(3), P(3).
- الموقع P(3) يدل على موقع النقطة التالية ((T(1), P(1)، حيث يشير الموقع P(1) إلى النقطة التالية ((T(7), P(7). فالقيمة المراد إضافتها يجب أن تقع بين المركزين P(1) و P(8).



- لنفترض أننا نرغب في إضافة الرقم 6، نقوم أولا بتحديد موقع العقدتين المتتاليتين والتي تحتويان على قيمتين أكبر وأصغر من القيمة 6.
- ثم نقوم بفحص أول موقع ((T(5), P(5) والذي يشير إلى موقع النقطة التالية (T(3), P(3)). الموقع P(3) يدل على موقع النقطة التالية ((T(1), P(1)، حيث يشير الموقع P(1) إلى القيمة المطلوبة 10.
- لحذف العقدة 10 يتم فقط تغيير قيمة المؤشر للعقدة السابقة، ووضع قيمة العقدة التالية، أي أن P(1)=7 تصبح P(1)=4، عندها سوف تصبح العقدة 10 كأنها غير موجودة أو تم مسحها.



3.3 تحقيق القوائم المترابطة Implementation of Linked lists

- إن تحقيق القوائم المترابطة برمجياً يجب أن يضمن تحقيق ثلاثة عمليات أساسية:
  - ترتيب المعلومات ضمن عقد.
  - الوصول إلى أية عقدة بالدلالة إلى مؤشرها.
  - خلق عقدة جديدة أو حذف عقدة غير مستخدمة.
- تحتوي لغات البرمجة على تعليمات يمكن أن تحقق العمليات الثلاث السابقة، ولكن إذا كانت هذه التعليمات غير متوفرة فانه يجب تصميم برامج جزئية تحققها.
- على المصمم أن يفكر أيضاً بالطريقة المناسبة لتخزين المعلومات، ويمكن في كثير من الأحيان استخدام مصفوفات، فإذا كانت لدينا k معلومة لكل حدث فإن المحاكي يحتاج إلى مصفوفة بطول k+1 لكل حدث، وإذا افترضنا أنه نحتاج إلى n حدث لتصميم نظام المحاكاة، فان حجم التخزين يحتاج إلى n مصفوفة بطول k+1.
- لتحقيق القوائم المترابطة بسهولة سوف نعتبر أن كل عقدة تتألف من حقلين فقط: الأول لتخزين المعلومات Data، والثاني لتخزين قيمة المؤشر Link. [8 إيضاح]
- سوف نستخدم برنامجين جزئيين للتحكم بفضاء التخزين وهما:
  - *Getnode*(X) سوف يضيف العقدة إلى القائمة المترابطة عند الموقع X.
  - *Ret*(X) والذي يعيد العقدة ذات القيمة X إلى الحالة الابتدائية، أي إلى فضاء التخزين

[9 إيضاح]

- سوف نشرح أربع خوارزميات تستخدم لتحقيق القوائم المترابطة:
  - A. خوارزمية إنشاء قوائم مترابطة Create linked lists.
  - B. خوارزمية حذف عقدة Deleting of a node.
  - C. خوارزمية إضافة عقدة جديدة Inserting of a new node.
  - D. خوارزمية التحكم بفضاء التخزين Managing the storage.

A – خوارزمية إنشاء قوائم مترابطة

- تقوم الخوارزمية التالية بإنشاء قائمة مترابطة مكونة من عقدتين تحتويان الرقمان 1 و 4، حيث يشير المتحول F إلى المؤشر لأول عقدة و I الموقع المحتمل للعقدة الثانية.

```
Procedure Create(F)

call Getnode(F)
Data(F)=1,
call Getnode(I)
Link(F)=I
Data(I)=4
Link(I)=0

End Create
```

B – خوارزمية حذف عقدة

- تقوم الخوارزمية التالية بحذف عقدة تحتوي على القيمة 10 من القائمة المترابطة المبينة في الفقرة السابقة حيث نفترض أن القيمة 10 تظهر مرة واحدة في القائمة، وباعتبار أن المتحول F يمثل المؤشر لأول عقدة و C=10 الموقع المحتمل للعقدة التي تحتوي على القيمة المطلوبة.

```
Procedure Delete (F,C)

If F=0 then error the list is empty
I=J=F
Do  {

   If Data(I)=C Then {
      Link(J)=Link(I)
      Ret(I)
      exit
   }

   Else {
     J=+I
     I= Link(I)
   }
}

End Delete
```

C – خوارزمية إضافة عقدة جديدة

- تقوم الخوارزمية التالية بإضافة عقدة جديدة تحتوي على القيمة C، وباعتبار أن المتحول F يمثل المؤشر لأول عقدة، والمتحول G يحتوي على موقع التخزين الفارغ المقترح.

```
Procedure Insert(F,G)

If F=0 then {
   Getnode(G)
   Data(G)=C
   Link(G)=0
   exit
}
If Data(F)≥C then {
   Getnode(G)
   Data(G)=C
   Link(G)=F
   F=G
   exit
}
J=F
I=Link(J)

do {
   if Data(I)≥C then {
      Getnode(G)
      Data(G)=C
      Link(G)=Link(I)
      Link(J)=X
      exit
   }
   else {
      J=I
      I=Link(I)
  }
}
End Insert
```

D – خوارزميات التحكم بفضاء التخزين

- تقوم الخوارزمية التالية بتهيئ فضاء التخزين لتحديد عدد العقد اللازمة لتحقيق القائمة المترابطة، حيث يمثل المتحول n عدد العقد المفترض استخدامها في القائمة، ويمثل المتحول v مؤشر أول عقدة.

```
Procedure Init(n)

For i=0 to n-1 do {
   Link(i)=i+1
}
 Link(n)=0
v=1
End Init
```

- تقوم الخوارزمية *getnode*(x) بإزالة العقدة من فضاء التخزين، أي تقوم بتهيئ عقدة جديدة، حيث يشير x إلى المؤشر المستخدم في القائمة.

```
Procedure Getnode(x)

If v=0 then error: no more nodes
else {
    x=v
    v=Link(v)
}
End Getnode
```

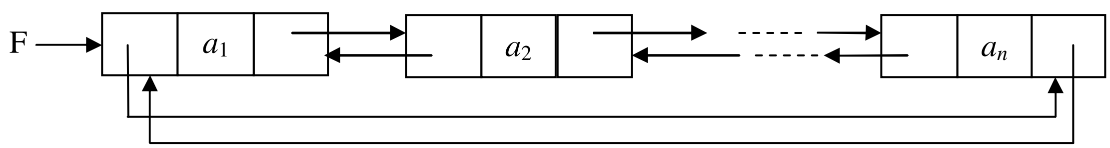
- تقوم الخوارزمية *Ret*(x) بإضافة عقدة فارغة إلى فضاء التخزين، أي تقوم بحذف عقدة من القائمة، حيث يشير x إلى المؤشر المستخدم في القائمة.

```
Procedure Ret(x)

Link(x)=v
v=x
End Ret
```

4.3 القوائم المترابطة الزوجية Double linked lists

- تعتمد طريقة القوائم المترابطة الزوجية على تخزين المعلومات عشوائياً ضمن أجزاء مختلفة في الذاكرة وعلى شكل عقد nodes كما هو الحال في القوائم المترابطة، ولكن يتم عملية الارتباط بين العقد في اتجاهين مختلفين.
- حيث تحتوي القائمة على مؤشرين عوضاً عن مؤشر واحد، يخصص كل واحد منهما للإشارة إلى اتجاه مختلف.



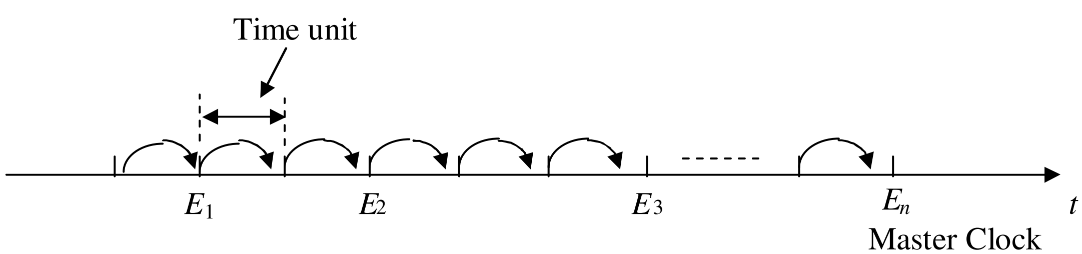
- تتميز هذه الطريقة بأنها تعطي مرونة أكبر عند تصميم نظام المحاكاة.

4. طريقة تقدم واحدة الزمن Unit time advanced

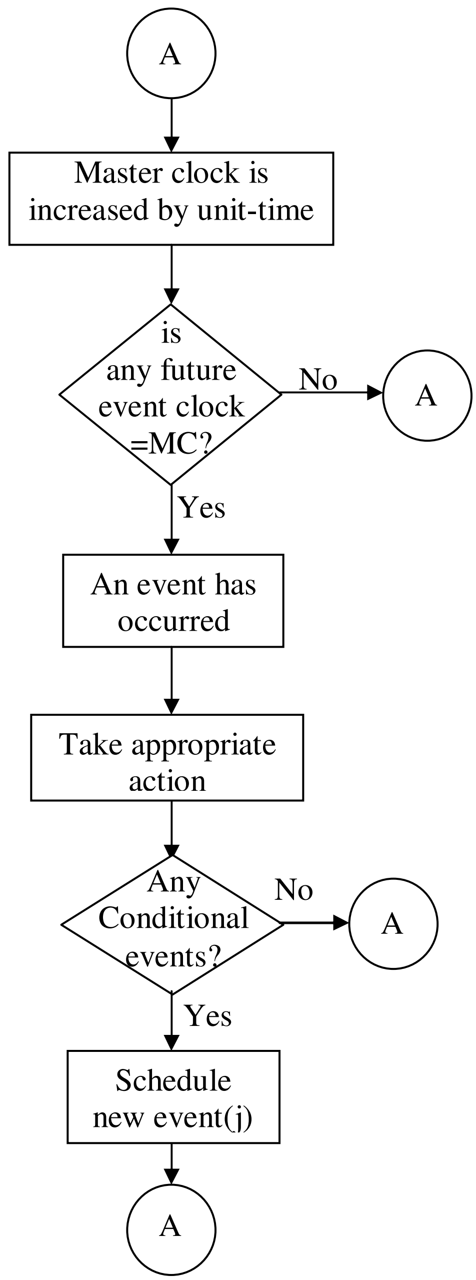
- تعتمد هذه الطريقة بشكل عام على زيادة الساعة الرئيسية Master Clock بقيمة ثابتة تساوي إلى واحدة الزمن one unit time. وتعتمد على الخطوات التالية:

1. في كل مرة تزاد فيها الساعة الرئيسية لمرة واحدة بمقدار واحدة الزمن.
2. يتم اختبار جميع أزمان الأحداث المستقبلية future event clocks.
3. عند تساوي الساعة الرئيسية مع أحد أزمان الأحداث المستقبلية لحدث ما ينفذ الحدث، ومن ثم نعود إلى الخطوة الأولى.
4. عند عدم التساوي نعود إلى الخطوة الأولى ويتم تكرار البحث.

- الشكل التالي يبين كيفية زيادة الساعة الرئيسية بواحدة الزمن



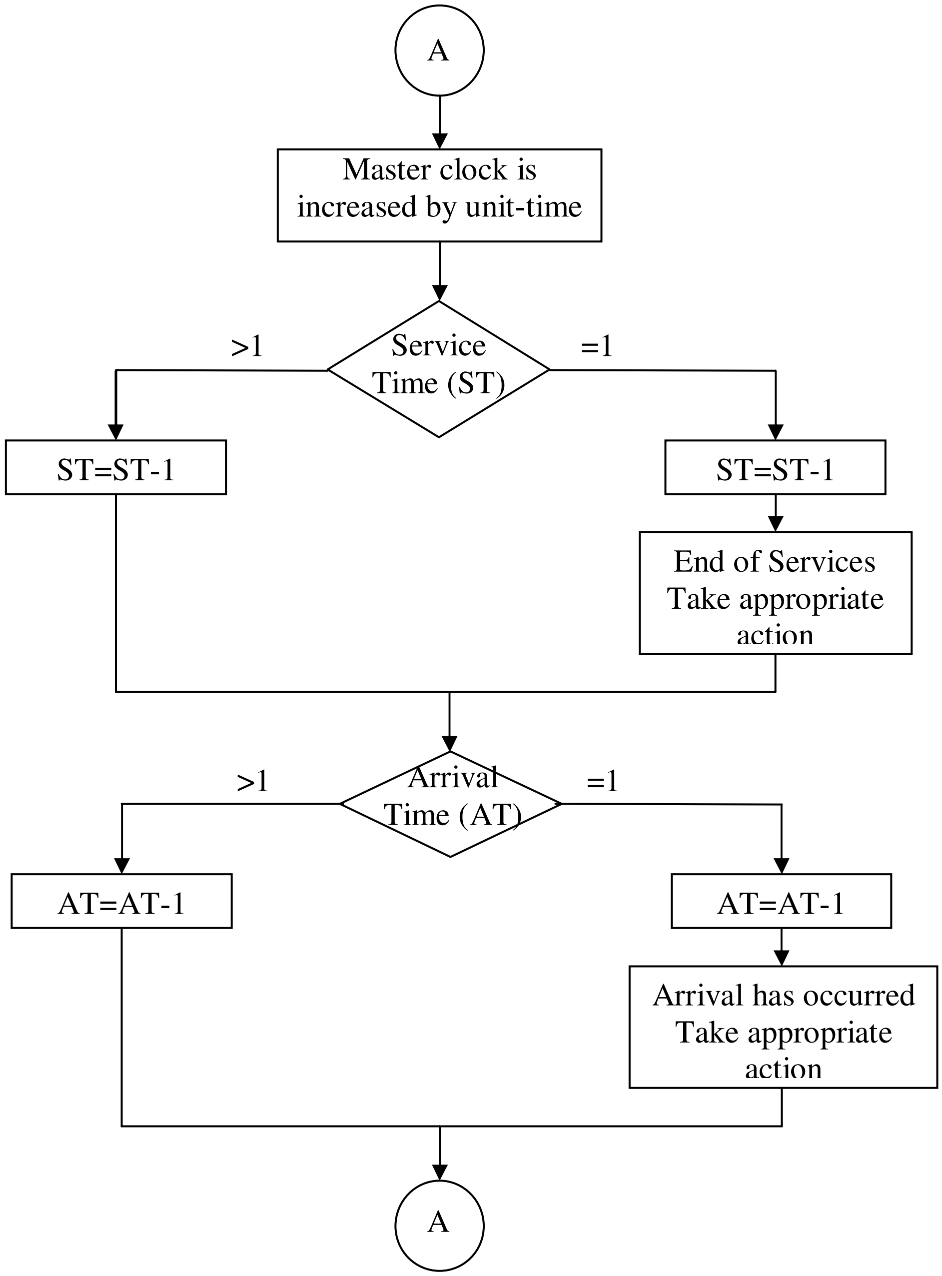
o يبين الشكل التالي المخطط التدفقي لهذه الطريقة.



- نلاحظ من المخطط انه لتحقيق المحاكاة بشكل صحيح يجب أن تتوفر لدينا مسبقا جميع أوقات حدوث الأحداث المستقبلية future event clocks، ولكن لا يمكن تحقيق ذلك من الناحية العملية بشكل دقيق، مثلاً لا يمكن توقع زمن إصلاح الآلة بشكل دقيق قبل البدء بإصلاحها، لذلك يجب اعتماد طريقة معدلة.
- الطريقة المعدلة تعتمد على الفترة الزمنية اللازمة لاستمرار الحدث بدلاً من وقت ظهور الحدث (زمن صلاح الآلة عوضاً عن وقت حدوث الإصلاح).
- في هذه الحالة تعدل الخوارزمية بحيث تصبح كالتالي:

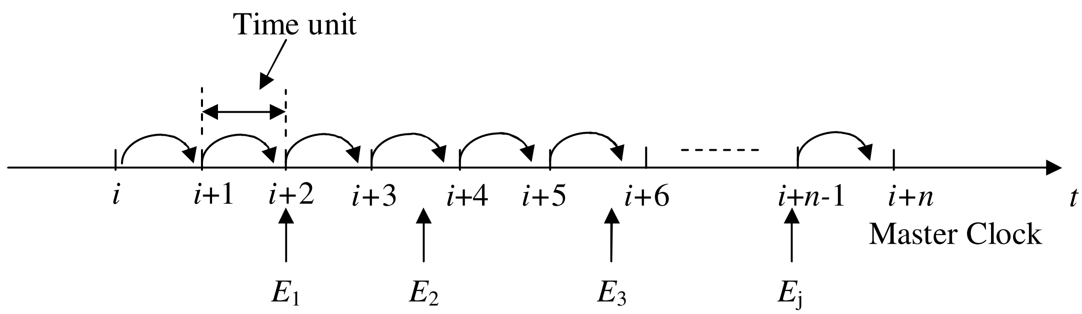
1. في كل مرة تزاد فيها الساعة الرئيسية لمرة واحدة بمقدار واحدة الزمن.
2. يتم إنقاص جميع أوقات الأحداث المستقبلية future event clocks، بمقدار قيمة واحدة الزمن.
3. عند تساوي أحد أوقات الأحداث المستقبلية إلى الصفر ينفذ الحدث، ومن ثم نعود إلى الخطوة الأولى.
4. عند عدم المساواة نعود للخطوة الأولى ويتم تكرار البحث.

- يبين الشكل التالي المخطط التدفقي للطريقة المعدلة وذلك عند محاكاة نظام المخدم في المصنع.



1.4. اختيار واحدة الزمن Selecting unit time

- عند اختيار قيمة واحدة الزمن يجب في البداية أن نحقق الشرط بان جميع الأحداث الممكن حصولها يجب أن تغطى بهذه القيمة، فمثلاً تحدث المشكلة عندما يكون زمن حدوث الحدث رقم حقيقي (كسري)، بينما تكون واحدة الزمن رقما صحيحاً، عندها سوف يظهر الحدث بين تغيرين زمنين (أي محصور ضمن مجال).



- لتفادي هذه المشكلة نستخدم إحدى الطرق التالية:
  - الأولى: نستخدم قيمة واحدة زمن صغيرة جداً بحيث تغطي كل الأحداث التي تظهر. هذه الطريقة سوف تؤدي إلى أن برنامج المحاكاة سوف يستغرق وقتاً طويلاً في البحث عن الحدث.
  - الثانية: بجعل قيمة واحدة الزمن نصف قيمة أقل متحول العشوائي مولد في نظام المحاكاة.
  - الثالثة: بتنفيذ المحاكاة عددت مرات مختلفة بقيم مختلفة لواحدة الزمن، مثلاً بالبدء بقيمة صغيرة ومن ثم زيادتها خطوة فخطوة بحيث يتم التحقق من أن جميع الأحداث قد تم تغطيتها.

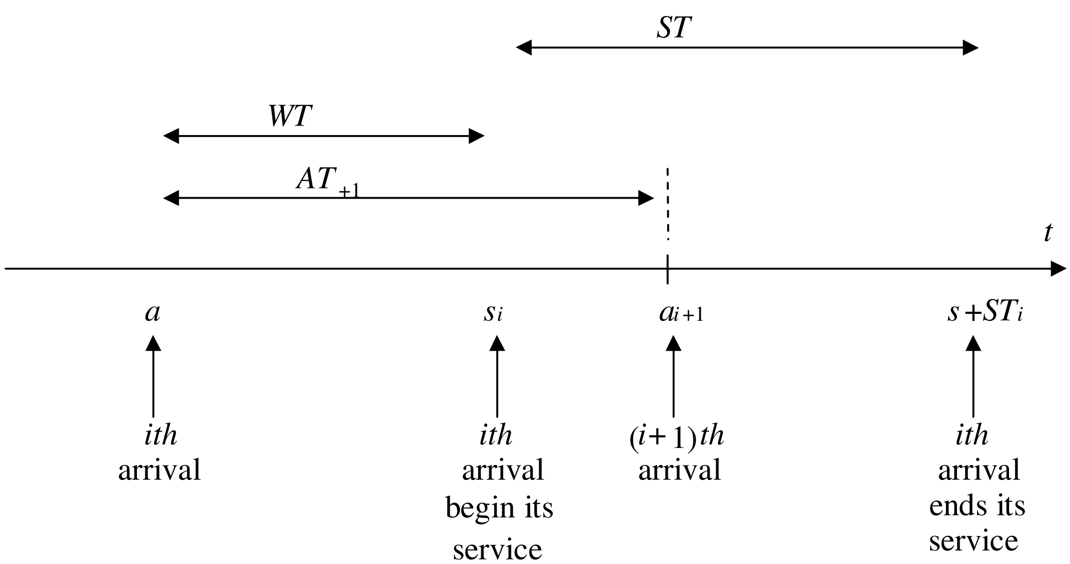
2.4. استخدامات طريقة تقدم واحدة الزمن، وطريقة تقدم الحدث

- يتم استخدام طريقة التقدم بواحدة الزمن عندما يكون نظام المحاكاة يتألف من عدد كبير من الأحداث المتقاربة من بعضها البعض، في هذه الحالة يتم اختيار الأحداث بسرعة و بسهولة عاليين، وتكون بأفضل حالتها عندما تكون الأحداث متباعدة عن بعضها البعض بقيمة تساوي واحدة الزمن.
- بينما يتم استخدام طريقة الحدث المتقدم عندما يحتوي نظام المحاكاة على عدد قليل من الأحداث المتباعدة عن بعضها البعض، حيث أن طريقة الزمن المتقدم سوف تأخذ وقتا طويلاً في البحث عن الحدث التالي.

5. طريقة محاكاة النشاط Activity based simulation design

- تعتمد هذه الطريقة في المحاكاة على مراقبة النشاط activity الذي يحدث أثناء المحاكاة بدلاً من مراقبة الحدث.
- وتستخدم هذه الطريقة بكثرة في محاكاة الأنظمة المعقدة complex systems حيث يصعب في هذه الأنظمة تحديد الأحداث المتلاحقة.
- على سبيل المثال في نظام المخدم في المصنع يوجد حدثين (تعطل الآلة أو إصلاحها)، بينما يمكن لهذا النظام اعتباره مجموعة من الأنشطة أو العمليات التالية:
  a) الدخول إلى الصف inter arriving (التعطل).
  b) الخروج من الصف being served (الإصلاح)
  c) الانتظار في الصف waiting for services.
- المهم معرفة بداية ونهاية كل نشاط من الأنشطة.

- لنفترض أننا نريد أن نقوم بتصميم نظام محاكاة بطريقة محاكاة النشاط لمخدم زبون وحيد single sever queuing، إذا :
  a) لنفترض أن زمن التخديم ST_i للزبون i.
  b) لنفترض أن زمن الإنتظار WT_i للزبون i
  c) لنفترض أن زمن الإنتظار بين الزبون   و الزبون التالي هو AT_{i+1}
  d) لنفترض أن الزبون i يظهر عند الزمن a ويبدأ التخديم عند الزمن s وينتهي التخديم عند الزمن s_i+ST

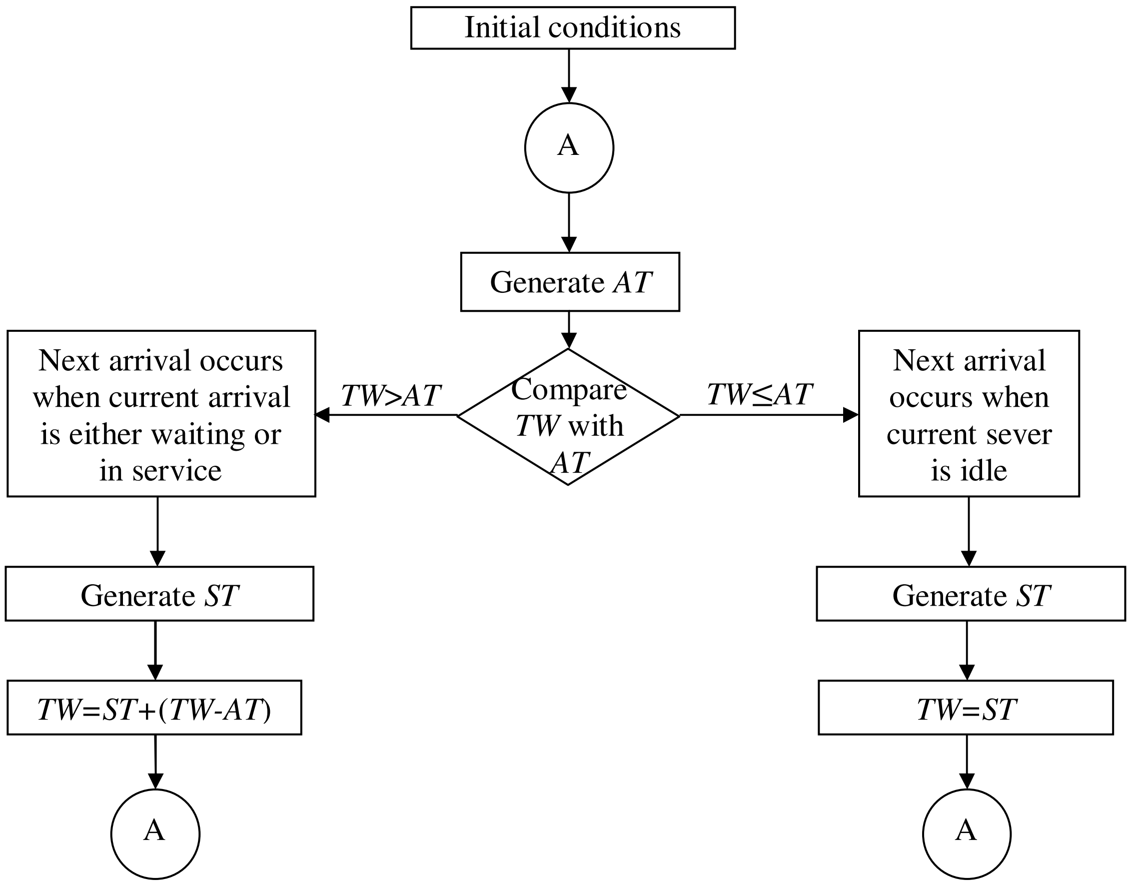


- لنفترض أننا نعرف قيم كل من WT_i و ST_i للزبون i، إذا لدينا ثلاث حالات:
  a) يظهر الزبون $i+1$ خلال فترة انتظار تخديم للزبون i .
  [10 إيضاح]
  b) يظهر الزبون $i+1$ خلال فترة تخديم للزبون i .
  [11 إيضاح]
  c) يظهر الزبون $i+1$ خلال بعد فترة تخديم للزبون i .
  [12 إيضاح]

- عندها يمكن حساب وقت الانتظار للزبون $i+1$ للحالات الثلاث السابقة على الترتيب:
  a) $WT_{i+1} = (WT_i - AT_{i+1}) + ST_i$.
  b) $WT_{i+1} = (WT_i + ST_i) - AT_{i+1}$.
  c) $WT_{i+1} = 0$

عند حساب WT_i نستطيع حساب جميع قيم الزبون $i+1$ ومنها يمكن إيجاد القيم الخاصة بالزبون التالي $i+2$ وهكذا..

o يبين الشكل التالي المخطط التدفقي لنظام المحاكاة باستخدام طريقة محاكاة النشاط.



6 أمثلة Examples

6. في هذه الفقرة سوف نشرح مثالين معتمدين على طرق المحاكاة المبينة سابقاً وهما:
   - نظام جرد مستودع Inventory system.
   - نظام الصف الدائري Round robin queue.

1.6 نظام جرد مستودعات Inventory system

7. يهدف نظام جرد المستودعات إلى إعطاء القرار الصحيح من أجل خفض الكلفة الاقتصادية لمنشئة صناعية أو تجارية إلى أقل كلفة ممكنة، ويكمن القرار الصحيح في شراء الكمية الصحيحة من البضائع وفي الوقت المناسب.
8. وتتعلق الكلفة الاقتصادية بثلاث عوامل أساسية هي:

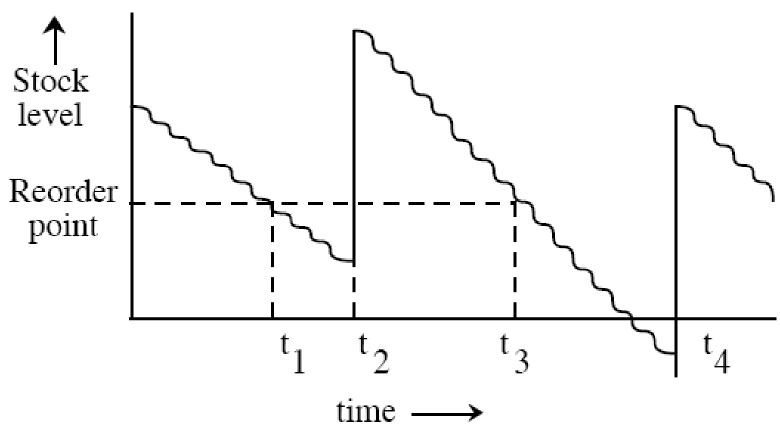
- كلفة تخزين البضائع، أي كلفة تخزين مادة واحدة في المستودع لمدة زمنية معينة وبدون استخدام. وتتعلق هذه الكلفة بمجموعة عوامل منها كلفة رأس المال المجمد مثلاً، كلفة تقادم البضاعة، أو كلفة تلف البضاعة نفسها مع الوقت.
- كلفة تشغيل البضائع، وتتعلق هذه الكلفة مثلاً بكلفة طلب البضاعة، وكلف الشحن، وكلف استخدام البضائع في الإنتاج.
- كلفة النقص في البضائع المعروضة، وتتعلق هذه الكلفة مثلاً بكلفة توقف الإنتاج في حال نقص مادة ما، أو الخسارة المحتملة في عدم بيع منتج الناقص، أو حتى في الكلفة الإضافية للشحن في حال طلب خاص للمادة الناقصة.

9. لنفترض الآن أن المتحول I_t يمثل كمية البضائع عند البدء أي في الزمن t، وأن S كمية المواد المضافة إلى النظام و D كمية المواد المطلوبة بين الفترة الزمنية t و t'، وبالتالي فأن كمية البضائع عند الزمن t' يساوي:

$$I_{t'} = I_t + S - D$$

عندما تصبح القيمة $I_{t'}$ تحت مستوى محدد عندها يجب إعطاء أمر بطلب المادة الناقصة.

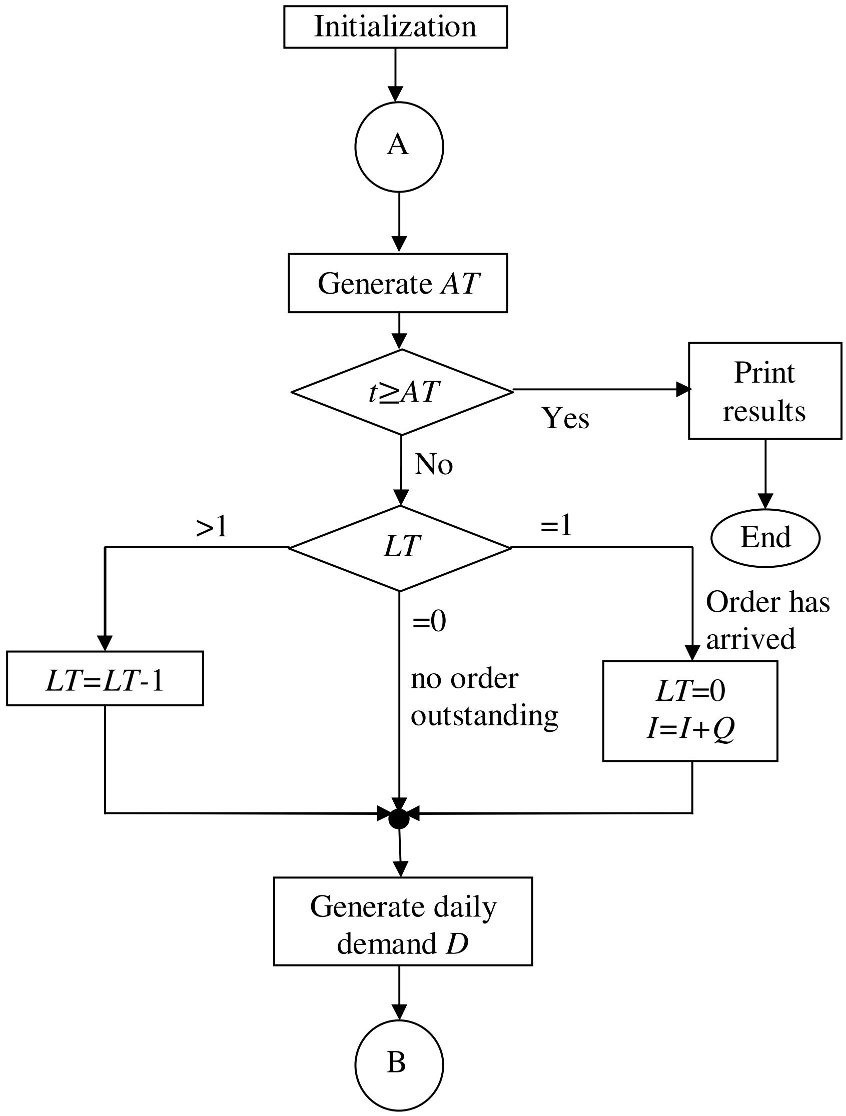
10. لنفترض أن طلب مادة (أو مواد) جديدة يتم يومياً وفي نهاية يوم العمل، وأن زمن التأخير lead time وهو الزمن اللازم لوصول الكمية المطلوبة إلى مستودعات الشركة (أي الفرق بين وقت طلب المادة و وقت وصول المادة إلى المستودع) سوف يحسب من اليوم التالي حيث أن الطلب يتم في نهاية يوم العمل. يمكن أن نلاحظ أن تغير قيمة البضائع في المستودع تتغير وفق الشكل التالي:

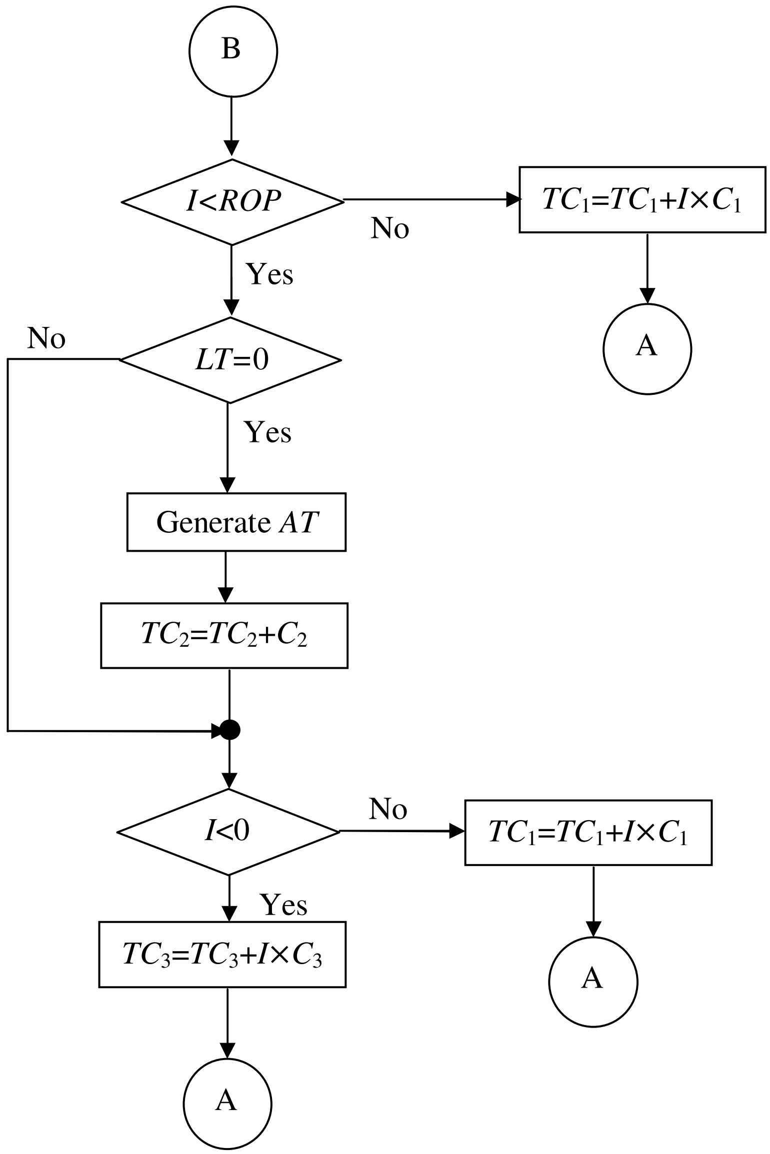


11. سوف نستخدم نظام المحاكاة بطريقة تقدم الزمن ونعتبر أن واحدة الزمن تساوي يوماً وحداً حيث أن الطلب يتم يومياً وفي نهاية العمل الرسمي.

- سوف نعتبر مجموعة مدخلات إلى النظام وهي:
- النقطة التي يتم عندها الطلب ROP.
- كمية البضاعة المطلوبة Q.
- كمية البضاعة عند البدء BI.
- الزمن الكلي T.
- الكلفة الاقتصادية للتخزين والكلفة الاقتصادية للطلب والكلفة الاقتصادية للنقص C_1، C_2، C_3 على الترتيب.
- الكلفة التقتصادية الكلية والمطلوب حسابها لكل من للتخزين والطلب والنقص TC_1، TC_2، TC_3 على الترتيب.

12. إن المخطط التدفقي للنظام مبين في الشكل التالي:

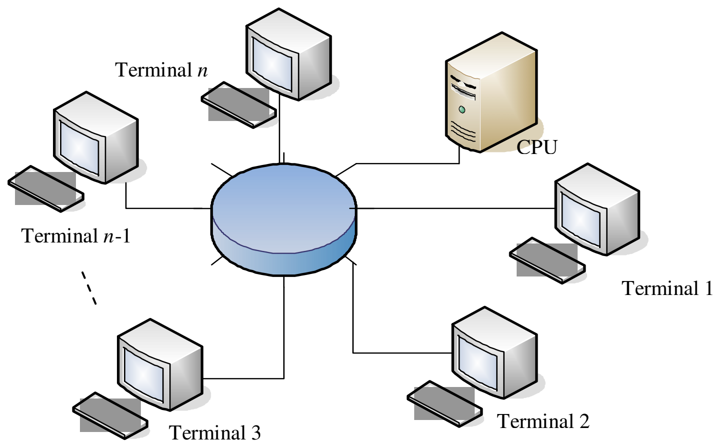

Initialization
A
Generate AT
t≥AT
Yes
Print results
End
No
LT
>1
=1
=0
Order has arrived
LT=LT-1
no order outstanding
LT=0
I=I+Q
Generate daily demand D
B



B
I<ROP
No
TC1=TC1+I×C1
A
Yes
No
LT=0
Yes
Generate AT
TC2=TC2+C2
I<0
No
TC1=TC1+I×C1
Yes
TC3=TC3+I×C3
A
A

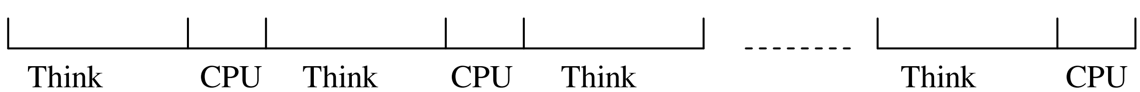
2.6 نظام الصف الدائري Round robin queue

13. يهدف نظام الصف الدائري إلى تصميم نظام محاكاة لدراسة أداء مجموعة N طرفية تتشارك جميعها بمعالج صغري CPU واحد.



14. لنفترض أن المستخدم عند كل طرفية يفكر لفترة زمنية ما Thinking (أي يقوم بكتابة سطر مثلاً، يفكر ماذا يفعل بعد ذلك..)، ومن ثم يستخدم المعالج المركزي أو يقوم بتنفيذ أمر ما عندما يقوم بالضغط على مفتاح Enter.

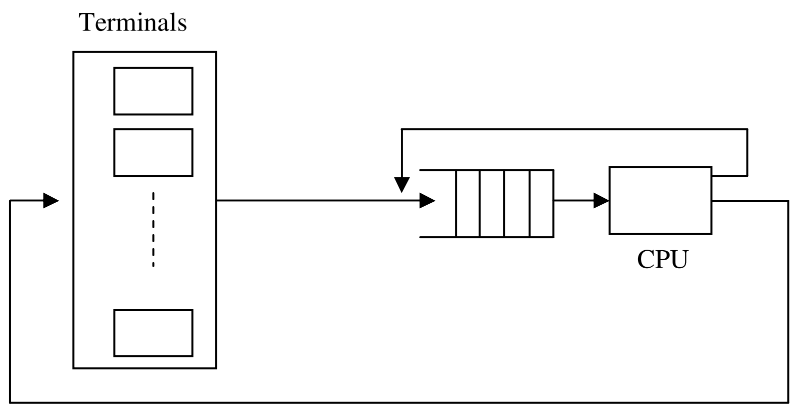
15. لنفترض أن المستخدم يقوم بالتفكير ومن ثم استخدام CPU بشكل متناوب ومتكرر، كما هو موضح بدورة العمل في الشكل التالي، حيث لا يمكن للمستخدم خلال استخدام المعالج أن يقوم بإرسال أمر آخر.



16. تعمل وحدة المعالجة CPU بالتناوب بين الطرفيات بشكل حلقة Round robin، بحيث تنفذ التعليمة في الطرفية الأولى ومن ثم الثانية، إلى أن تنتهي بأخر طرفية وعندها تعود الطرفية الأولى لتخدمها من جديد.

17. يخصص المعالج CPU زمن التنفيذ صغيراً جدا لكل طرفية يسمى Quantum، وبحيث إذا لم ينتهي المعالج CPU من إنهاء العمل المخصص للطرفية خلال الوقت Quantum ينقل بقية العمل لهذه الطرفية إلى بداية الصف، ويتابع CPU تنفيذ العمل للطرفية اللاحقة.

18. يبين الشكل التالي الفكرة الأساسية من دورة العمل لوحدة المعالجة CPU.



19. في الحالة العملية يكون زمن التفكير أطول بكثير من زمن التنفيذ، فمثلاً زمن التفكير يكون بحدود 30 ثانية، ويكون زمن التنفيذ Quantum بحدود 1msec. لنفترض أن تنفيذ طلب ما يحتاج إلى 5 sec من وحدة المعالجة CPU إذا يتطلب تنفيذ هذا العمل 5000 دورة ضمن الصف.

20. يشبه نظام المحاكاة نظام محاكاة لإصلاح الآلات في مصنع، حيث يشبه CPU عمل عامل الإصلاح، ويختلفان بأن الدخول إلى الصف يتم بشكل دائري وليس بطريقة FIFO.

21. ويمكن تقسيم نظام المحاكاة إلى حدثين أساسيين:

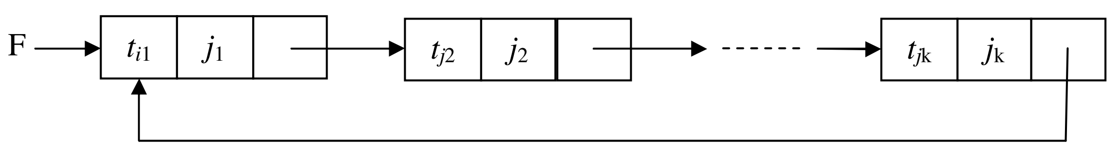
- الأول: هو وصول العمل إلى صف الانتظار للمعالج، وقسم إلى قسمين:
  - i. وصول عمل جديد ناتج عن طلب من أحد الطرفيات (عند الضغط على مفتاح Enter مثلاً)،
  - ii. أو وصول من عمل غير منتهي.
- الثاني: خروج العمل من الصف. ويقسم أيضاً إلى قسمين
  - i. الأول عند انتهاء من تنفيذ العمل المطلوب،
  - ii. والثاني هو انتهاء الوقت المخصص Quantum لتنفيذه لاحقاً.

22. يمكن تصميم نظام محاكي للصف الدائري باستخدام طريقة الحدث المتقدم.

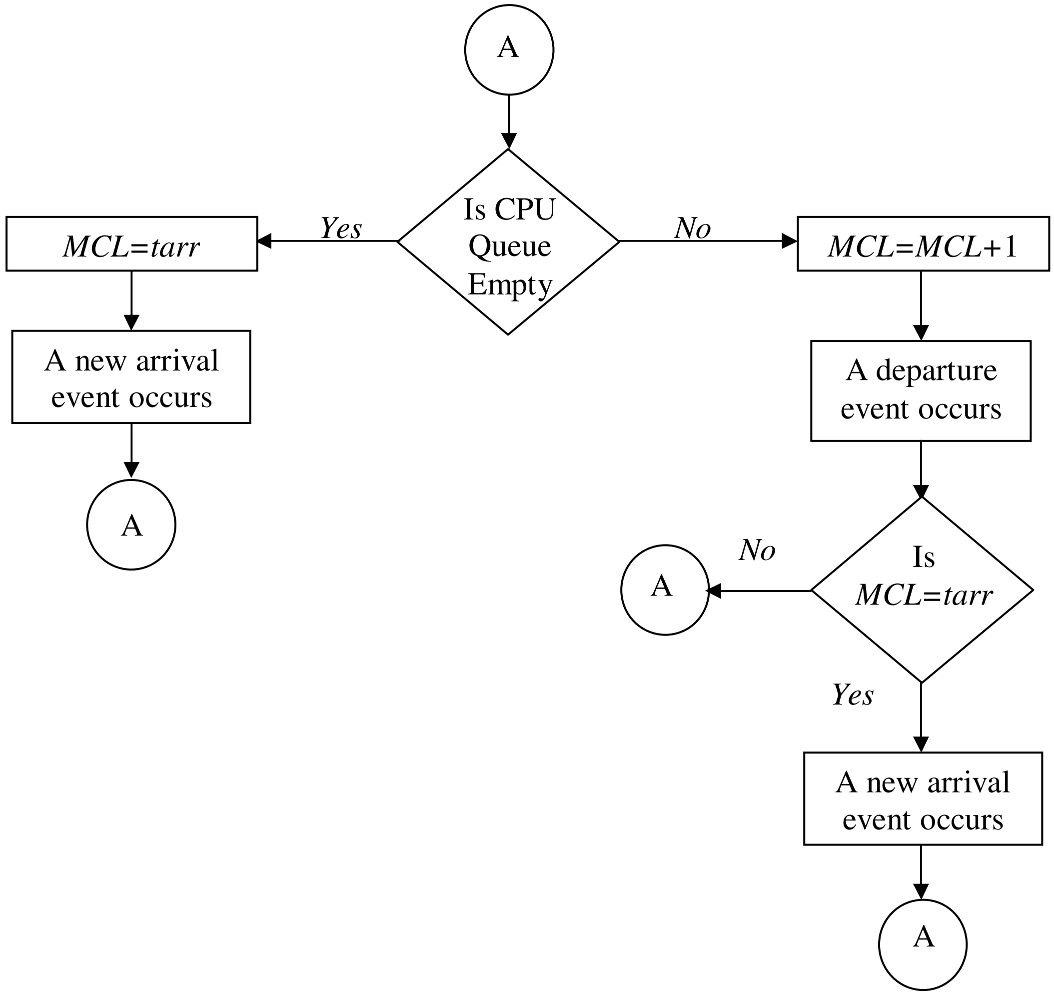
23. سوف تحتوي قائمة الأحداث المستقبلية على حدث خروج العمل من الصف CPU و دخول الأحداث الجديدة إلى الصف، وباعتبار أن زمن التنفيذ ثابت ويساوي Quantum فان القائمة ثابتة.

24. ولتصميم القائمة سوف تحتوي كل عقدة نوعين من المعلومات الأول الحدث التالي Future event، والثاني انتهاء العمل (للدلالة إذا كان العمل المراد تنفيذه قد نفذ كاملاً أو يحتاج إلى إعادته إلى بداية القائمة).

25. ويبين الشكل التالي قائمة الأحداث المستقبلية حيث t_j يمثل زمن دخول الحدث إلى القائمة.



26. يبين الشكل التالي المخطط التدفقي لمحاكي الصف الدائري:



ملحقات

[1 إيضاح]

راجع مثال إصلاح الآلات في مصنع المبين في الفصل الأول.

[2 إيضاح]

إذا يكفي ملاحظة أو تسجيل تغير الأحداث events في النظام لإجراء المحاكاة.

[3 إيضاح]

وذلك باستخدام المسجلة في عداد الساعة clock لكل حدث.

[4 إيضاح]

لذلك تدعى هذه الطريقة بطريقة تقدم الحدث حيث يتم الاعتماد على تغيرات الحدث في تصميم نظام المحاكاة.

[1 مثال]

لقد بينا في مثال إصلاح الآلات في المصنع (راجع الفصل الأول) انه يوجد حدثين فقط، الأول دخول الآلة إلى صف الإصلاح (أي تعطل الآلة) والثاني خروج الآلة من الصف (أي إصلاحها). حيث تدعى هذه الأحداث بالأحداث الأولية Primary events.

[2 مثال]

إن تكرار حدوث أعطال في الآلات ودخول الآلات إلى صف الانتظار قد يؤدي إلى حدوث حالة جديدة لا تخرج الآلات من صف الانتظار (إي لا تصلح) وذلك عندما يكون عامل الإصلاح بحاجة إلى وقت للراحة idle حتى ولديه عدد كبير من الآلات بحاجة إلى إصلاح.

[5 إيضاح]

سوف نشرح قائمة الأحداث المستقبلية في الفقرة التالية.

[6 إيضاح]

طبيعة الحدث يعبر عن الفعل الذي يجب اتخاذه عندما يظهر الحدث، أي البرنامج الجزئي المخصص لتنفيذ هذا الحدث.

[3 مثال]

لقد بينا في المثال إصلاح الآلات في المصنع (راجع الفصل الأول) انه يوجد حدثين فقط، الأول دخول الآلة إلى صف الإصلاح (أي تعطل الآلة) والثاني خروج الآلة من الصف (أي إصلاحها). ولكن عند الرغبة في تصميم نظام محاكاة لأنظمة معقدة complex systems فان عدد الأحداث يصبح كبيراً جداً، وبالتالي البحث عن حدث جديد سوف يحتاج إلى خوارزميات معقدة لتحقيق ذلك لتحقيق المطلوب بأقل عدد ممكن من العمليات الحسابية. و يعتمد كفاءة هذه الخوارزميات على كمية المعلومات التي وسوف تبقى في القائمة المستقبلية وكيفية تخزين هذه المعلومات.

[7 إيضاح]

نلاحظ أن كل عقدة تمثل بمربعين الأول يمثل المعلومات المخزنة في العقدة، حيث يمكن أن تكون هذه المعلومات متحول ما بسيط ويمكن أن تكون مجموعة مصفوفات، بينما يمثل المربع الثاني موقع العقدة الثانية أو اسم العقدة الثانية في القائمة. ونلاحظ أن المؤشر F يشير إلى أول عقدة بينما يشير المؤشر في العقدة الأخيرة إلى القيمة 0 مما يدل على أنه لا توجد عقدة تالية.

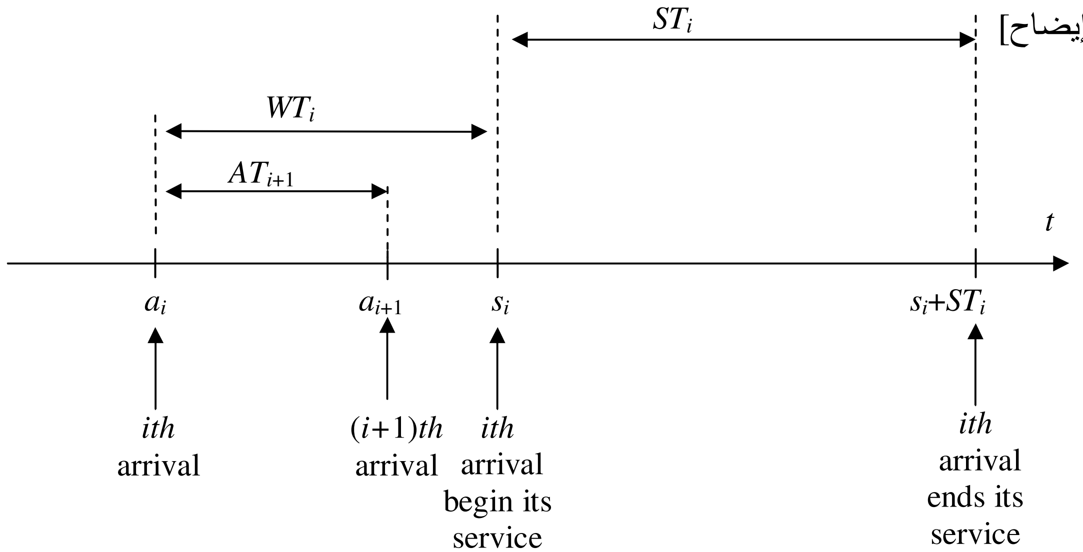
[8 إيضاح]

ويتم الولوج إلى المعلومة للعقدة i باستخدام Data(i) أو Link(i).

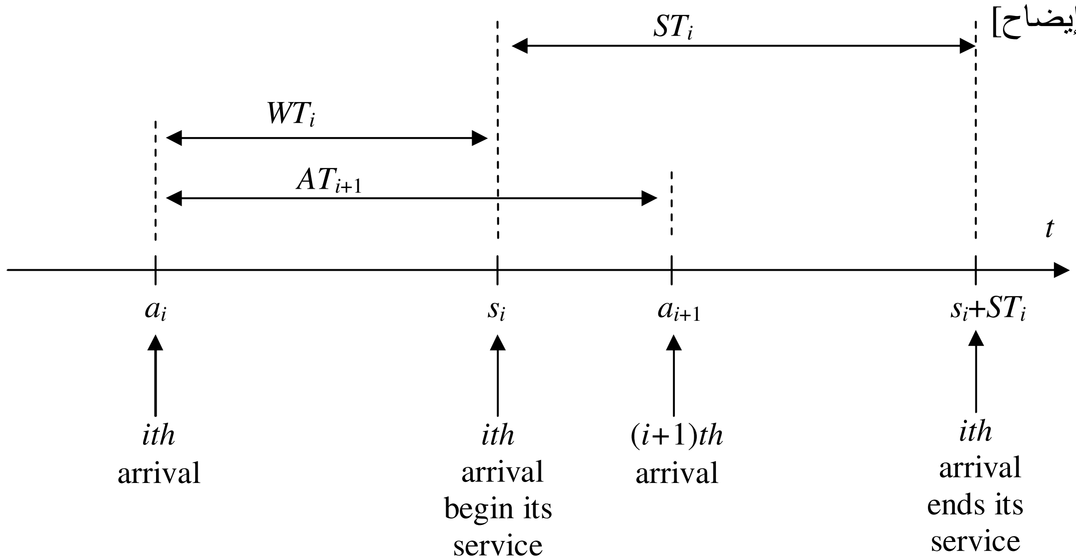
[9 إيضاح]

سوف نبين بالتفصيل هذين البرنامجين لاحقاً.

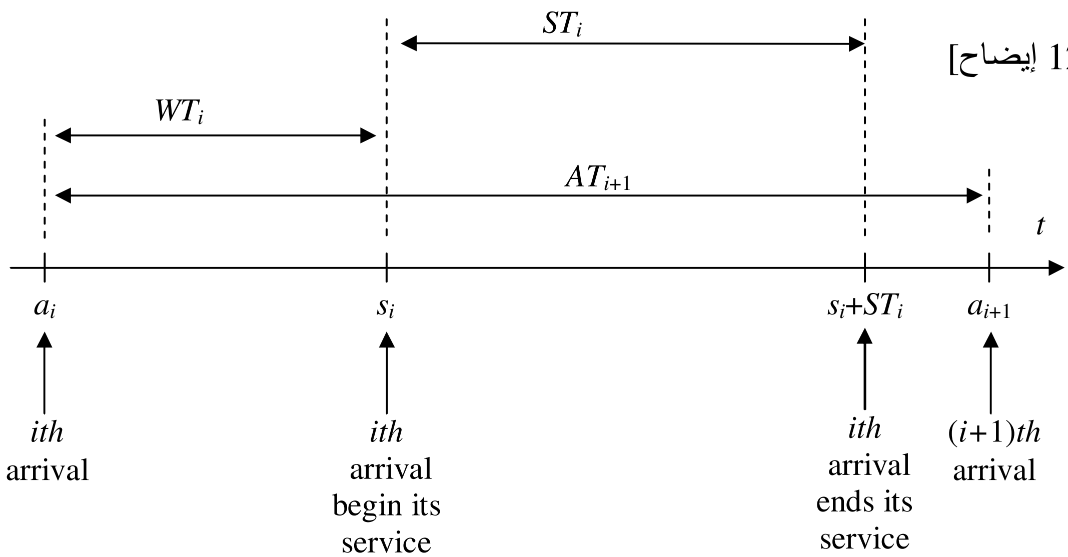
[إيضاح 10]



[إيضاح 11]



[إيضاح 12]



التمارين

ليكن لدينا الأنظمة التالية:

14- التحصيل (صناديق المحاسبة) في سوبر ماركت.

15- نظام المخدم في مصرف.

16- نظام مصاعد في بناء.

17- نظام إشارات مرور عند تقاطعات طرق.

18- غرفة انتظار في عيادة.

19- محطة محروقات.

20- كراج طابقي لانتظار السيارات.

21- مدرج طائرات في مطار.

22- نظام طاقة شمسية في منزل.

المطلوب:

أختر واحد من الأمثلة السابقة، ثم قم بشرح مبدأ عمل النظام واضعاً افتراضاتك وبحيث لا تؤثر على عمل النظام. ثم اكتب المخطط التدفقي للنظام في حال:

1- نظام تقدم الحدث.

2- نظام محاكاة النشاط.

مسائل

10- قم بتعديل برنامج المحاكاة الذي قمت بتصميمه سابقاً حول محاكاة مصنع، بحيث تضيف قائمة مترابطة، وأطبع النتائج.

11- كرر السؤال السابق عند زيادة عدد عمال الإصلاح من ٢ إلى ١٠ عمال وقارن النتائج.

12- قم بتنفيذ برنامج يقوم بمحاكاة نظام الصف الدائري المبين في الفقرة 2.6.

تقنيات التقييم لمخرجات نظم المحاكاة

Estimation Technology For Analyzing Endogenously Created Data

1. مقدمة

- حتى الآن استعرضنا في الفصول السابقة الطرق المستخدمة في بناء محاكي، حيث لاحظنا أن البناء يتمركز حول توليد أعداد عشوائية المرغوبة وذلك لمحاكاة ما هو موجود في الواقع.
- في هذا الفصل سنبين الطرق المستخدمة لتحصيل النتائج أو مخرجات النظام Endogenously، بالإضافة إلى الطرق المستخدمة لتحليل هذه النتائج.

2. تحصيل المخرجات Collecting endogenously created data

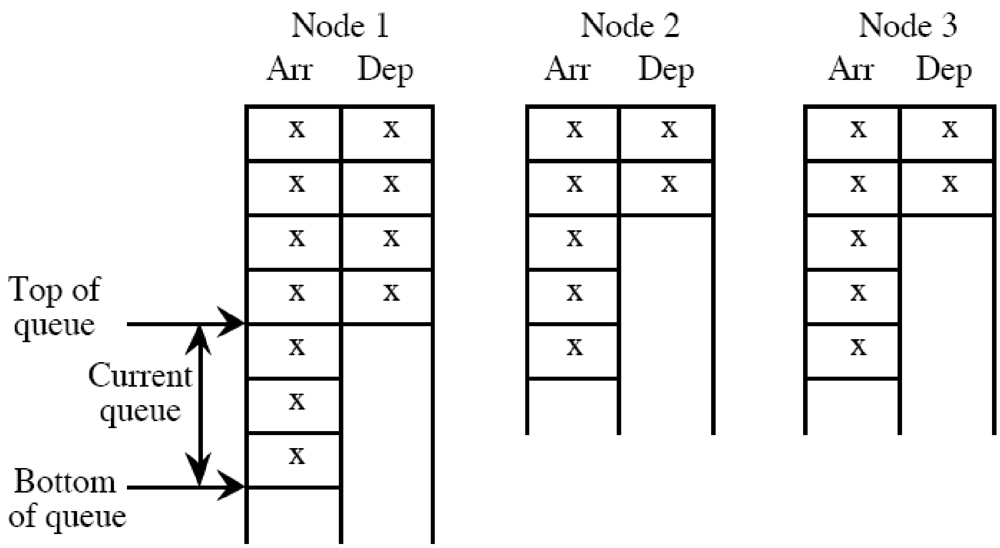
- يمكن اعتبار نظام محاكاة على أنه بناء لنظام ما تحت الاختبار، لتحقيق هذا الاختبار يجب على المصمم تحصيل متغيرات النظام الضرورية ليتمكن بعد ذلك من اختبار النظام المدروس.
- بالعودة إلى النظم المدروسة سابقاً يمكن تسجيل تغيرات الحدث للحصول على معلومات عن النظام وتغيراته مع الزمن، ولكن يجب في كثير من الحالات مراقبة معلومات أخرى ذات أهمية، [1 مثال] ، تسمى هذه المعلومات بالمخرجات endogenously.
- يمكن جمع المعلومات أو المخرجات باستخدام طرق متعددة، فمثلاً يمكن استخدام متحولات في البرنامج تستخدم كعدادات counters أو مسجلات لحفظ وتسجيل المخرجات

  [2 مثال] . [1 إيضاح]
- من المعلومات الممكن تحصيلها والمهمة تابع توزيع الاحتمالي للآلات المعطلة، والذي يمثل فعلاً عدد مرات تعطل الآلة خلال فترة العمل المدروسة (مثلاً خلال عام).
- وبافتراض انه لدينا n آلة فإننا نحتاج لتخزين المخرجات إلى مصفوفة من $n+1$ عنصر، حيث سوف يحتوي العنصر $a(i)$ من المصفوفة عدد مرات تعطل الآلة رقم i وعندما تتعطل أية آلة فان العنصر الموافق لها في المصفوفة سوف يزاد بمقدار واحد. عند انتهاء المحاكاة فان تابع التوزيع الكثافة الاحتمالي $p(n)$ يساوي إلى:

$$p(i) = \frac{a(i)}{T}, \quad i = 0,...,n$$

حيث T زمن المحاكاة الكلي.

o لنفترض أننا نريد دراسة أداء شبكة حاسوبية LAN عن طريق إجراء محاكاة للشبكة، فلتحسين أداء المحاكي يضاف لكل عقدة مصفوفة مزدوجة، كما في الشكل التالي:



o يخزن في العامود الأول من المصفوفة زمن وصول الرزمة packet إلى العقدة ويخزن في العامود الثاني زمن خروجها، فعند وصول الرزمة إلى العقدة يتم تخزين زمن الوصول في أول مكان فارغ في المصفوفة، وعند خروج الرزمة يتم تسجيل الزمن في المكان المقابل في العامود الثاني.

o إذا يمكن مراقبة حجم الازدحام congestion في عقدة ما بتحليل المعلومات المتوفرة في هذه المصفوفات. [2 إيضاح]

3. الحالات المستقرة والعابرة Transient states & steady states simulation

o يمكن في الحالة العامة يمكن تحصيل المعلومات ودراسة مخرجات نظام محاكاة في حالتين إما في الحالات المستقرة steady states أو الحالات العابرة transient states.

o لنفترض أننا نريد أن نجري إحصاء حول عدد الآلات المعطلة في مصنع، يجب قبل بدء المحاكاة تحديد الشروط الابتدائية initial condition مثلاً نعتبر أن عدد الآلات المعطلة مساوية للصفر، [3 إيضاح] [4 إيضاح] .

o خلال فترة الزمنية الصغيرة T يكون نظام المحاكاة في الحالة العابرة، وبعد الزمن T يصبح المحاكاة إنه في الحالة المستقرة.

1.3 الحالات العابرة في المحاكاة Transient state simulation

- تدرس الحالات العابرة في معظم الأحيان عند تغيير الشروط الابتدائية لنظام ما، أو عند وضع شروط ابتدائية لنظام محاكاة جديد. فمثلاً عند دراسة حالة مصنع عند افتتاحه لأول مرة.
- وتدرس أيضاً الحالات العابرة عندما لا يكون للنظام حالات مستقرة، فمثلاُ عندما يكون النظام في حالة التجارب ودائما متحولات النظام تتغير لتحسين أداءه.

2.3 الحالات المستقرة في المحاكاة Steady state simulation

- يتم تحليل ودراسة مخرجات نظام محاكاة في معظم الأحيان في الحالات المستقرة، عندها يجب تنفيذ المحاكي لمدة زمنية طويلة كافية ليستقر خلالها النظام.
- يتم اعتماد طريقتين:
  - ✓ الأولى هو البدء بنظام فارغ empty system أي لا توجد أية فعاليات قبل البدء بالمحاكاة .
  - ✓ الثانية هو البدء بشروط أولية تعطي حالة اقرب للحالة التي يمكن أن نحصل عليها عندما يستقر النظام.

[5 إيضاح]

- يجب عند تحصيل المخرجات التخلص من المعلومات الخاصة بالحالة العابرة لأنها لا تعطي انطباعاً صحيحاً على النظام، ولأنها معتمدة على الشروط الابتدائية.
- للتخلص من المخرجات الخاصة بالحالة العابرة يمكن استخدام إحدى الطريقتين التاليتين:
  - ✓ تشغيل المحاكي لفترة زمنية طويلة وبالتالي فان كمية المخرجات المحصلة والتي تتعلق بالحالة العابرة سوف يكون تأثيرها صغير جداً ومهمل.
  - ✓ يتوقف تحصيل المعلومات خلال الفترة العابرة، ويتم ذلك بتشغيل المحاكي حتى يصل إلى الحالة المستقرة وعندها يبدأ بتحصيل المعلومات من النظام لفترة زمنية بحيث تكفي لتحصيل كمية كافية من المعلومات.
- المشكلة في الطريقة الثانية هو في كيفية معرفة ما إذا كان النظام قد وصل إلى الحالة المستقرة أم لا. ويمكن ذلك بتشغيل المحاكي خلال فترات زمنية مختلفة ومتصاعدة $T_1>T_2>T_3>,\ldots>T_i$ ويتم مقارنة المخرجات بين كل فترة زمنية وفترة زمنية أخرى، فعندما تصبح تغيرات المخرجات صغيرة بين فترة زمنية وأخرى يصبح النظام بحالة الاستقرار.

4. تقنيات تقييم الحالات المستقرة Estimating techniques for SS

- يقصد بتقييم الحالات المستقرة، مجموعة القياسات التي يمكن أن تجرى على تابع التوزيع الاحتمالي لمتحول مخرجات النظام، حيث يتم مثلاً حساب المتوسط mean، الانحراف المعياري standard deviation، النسب المئوية للنتائج أو المخرجات.
- فمثلاً المعلومة المهمة في المصنع هو تابع التوزع الاحتمالي لتعطل الآلة، أو تابع التوزيع الاحتمالي لتعطل الآلات جميعها. كما أن نسبة الآلات المعطلة، أو نسبة زمن التعطل لآلة ما بالنسبة لوقت العمل مهم جداً لاتخاذ قرار مناسب من مدراء المعمل.

1.4. تقييم مجال الثقة لمتوسط المتحول العشوائي

- ليكن لدينا تتابع المخرجات x_1، x_2، x_3،....x_n نظام محاكاة، يتم حساب متوسط المتحول العشوائي يعطى بالعلاقة:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- للحصول على مجال الثقة للمتوسط يجب أولاً حساب الانحراف المعياري للمتحول العشوائي باستخدام العلاقة:

$$s_d^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n}(x_i - x)^2\right)$$

- أو مشكل مختصر:

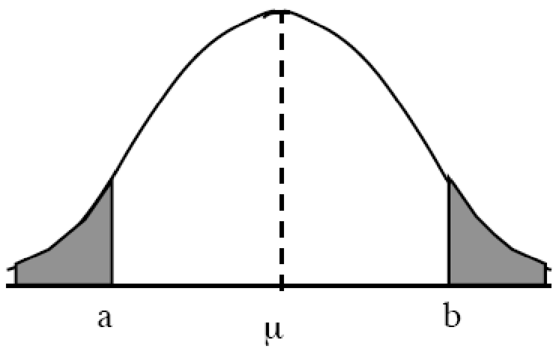
$$s_d^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{j=1}^{n} x_j\right)^2}{n}\right)$$

o وبالتالي فان مجال الثقة يعطى بالعلاقة:

$$\left(\overline{x} - 1.96\frac{s_d}{\sqrt{n}}, \overline{x} + 1.96\frac{s_d}{\sqrt{n}}\right)$$

عند 95% ثقة. [6 إيضاح]

o يمكن حساب المتوسط من أجل أية قيمة لمجال الثقة، إن معظم القيم المستخدمة هي 99%، 95%، و 90%، ويمكن التعبير عن مجال الثقة بأنه المجال المحصور بين القيمتين a و b في الشكل التالي:



[7 إيضاح]

o يمكن أيضاً التحقق ما إذا كانت المخرجات x_1، x_2، x_3،....x_n مترابطة مع بعضها البعض، فمثلاً توقف آلة ما يمكن أن يتعلق بتوقف آلة أخرى تعتمد عليها.

o إذا لا يمكن حساب قيمة المتوسط باستخدام المعادلة:

$$x = \frac{1}{n}\sum_{i=1}^{n} x_i$$

o إذاً يجب التحقق من ترابط المخرجات أولاً، ويتم ذلك سوف نستخدم الطرق التالية:

a) حساب تابع الترابط الذاتي autocorrelation function.

b) متوسط الحزم batch mean.

c) طريقة التكرار replication.

d) طريقة التجديد regenerative method.

a) حساب تابع الترابط الذاتي autocorrelation function

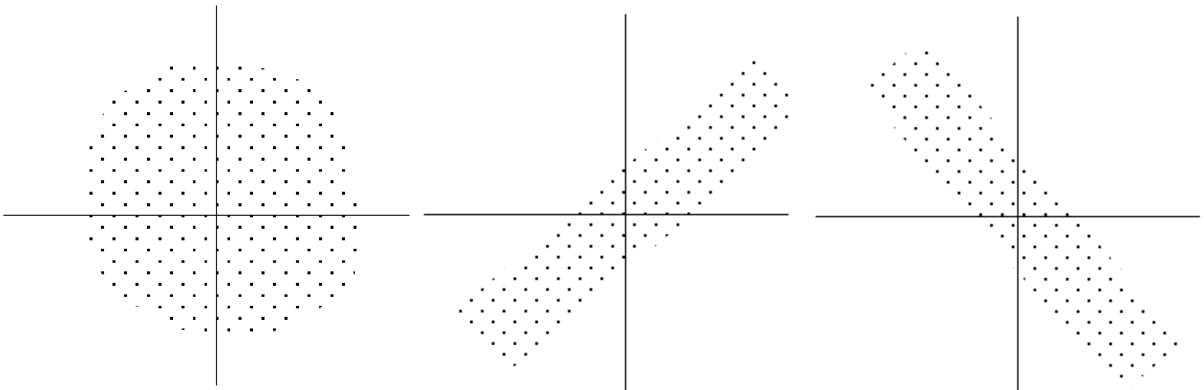
- ليكن X و Y متحولين عشوائيين، وليكن μ_x و μ_y المتوسط المتوقع expectation لهما على الترتيب، وليكن لدينا σ^2_x و σ^2_y تباين كل من المتحولين X و Y على الترتيب. يمكن حساب الترابط المتغاير covariance باستخدام العلاقة:

$$c_{xy} = E[(X-\mu_x)(Y-\mu_y)]$$
$$= E[XY] - \mu_x \mu_y$$

[8 إيضاح] ، حيث $E(X)$ المتوسط المتوقع للمتحول X.

- وتعبر c_{xy} على الترابط بين X و Y، فإذا كانت:
  - $c_{xy} = 0$ فان المتحولين X و Y غير مترابطين،
  - $c_{xy} > 0$ فان المتحولين X و Y مترابطين بشكل موجب،
  - $c_{xy} < 0$ فان المتحولين X و Y مترابطين بشكل سالب.
  - $c_{xy} = \sigma_x^2$ فان المتحولين X و Y متطابقين.

o يبين الشكل التالي حالات الترابط الثلاث.



o لتفادي حساب قيم لانهائية يمكن حساب الترابط correlation للمتحولين العشوائيين X و Y باستخدام العلاقة:

$$\rho_{xy} = \frac{c_{xy}}{\sigma_x \sigma_y}$$

o وتعبر ρ_{xy} على الاستقلالية بين X و Y، وتكون $-1 \leq \rho_{xy} \leq 1$ فإذا كانت:

- $\rho_{xy} = 0$ فان المتحولين X و Y مستقلين أي غير مترابطين،
- وإذا كانت ρ_{xy} قريبة من 1 فان المتحولين X و Y مترابطين بشكل موجب،
- وإذا كانت ρ_{xy} قريبة من -1 فان المتحولين X و Y مترابطين بشكل سالب،

- ليكن لدينا المخرجات x_1، x_2، x_3،....x_n والتي يمكن جمعها ضمن أزواج من النقاط (x_1,x_2)، (x_3,x_4)، (x_5,x_6)، .....(x_i,x_{i+1})،.... (x_{n-1},x_n). ولنعتبر زوج النقاط الأول هو من متحولات X والثاني Y وهكذا...، يسمى التابع ρ_{xy} بتابع الترابط الذاتي autocorrelation function ويساوي إلى:

$$r_1 = \frac{\sum_{i=1}^{n-1}(x_i - \overline{X})(x_{i+1} - \overline{Y})}{\sqrt{\sum_{i=1}^{n-1}(x_i - \overline{X})^2 \sum_{i=1}^{n-1}(x_{i+1} - \overline{Y})^2}}$$

حيث: $\overline{X} = \frac{1}{n-1}\sum_{i=1}^{n-1} x_i$ و $\overline{Y} = \frac{1}{n-1}\sum_{i=2}^{n} x_i$

- من أجل القيم الكبيرة لـ n يمكن تقريب العلاقة السابقة بحيث تصبح:

$$r_1 = \frac{\sum_{i=1}^{n-1}(x_i - \overline{X})(x_{i+1} - \overline{X})}{\sum_{i=1}^{n-1}(x_i - \overline{X})^2}$$

حيث $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$

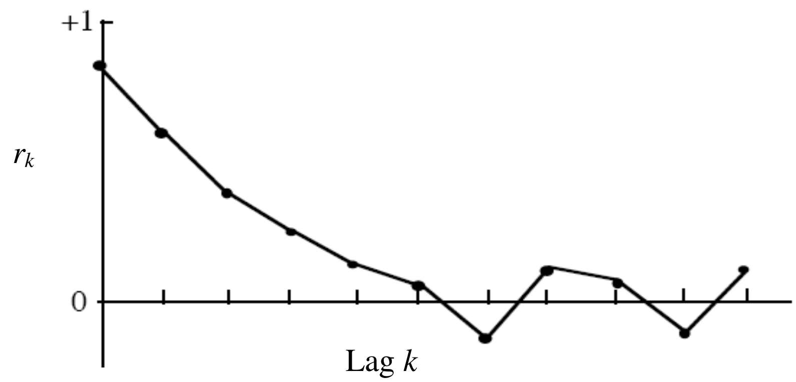
- يسمى تابع الترابط الذاتي r_1 تابع الترابط لخطوة واحدة lag 1 وهو يهتم بترابط عينتين متجاورتين.
- يمكن حساب تابع الترابط الذاتي للخطوة k باستخدام العلاقة:

$$r_k = \frac{\sum_{i=1}^{n-k}(x_i - \overline{X})(x_{i+k} - \overline{X})}{\sum_{i=1}^{n-1}(x_i - \overline{X})^2} = \frac{R_k}{R_0}$$

حيث R_x تابع الترابط المتغاير الذاتي والذي يساوي:

$$R_k = \sum_{i=1}^{n-k}(x_i - \overline{X})(x_{i+k} - \overline{X})$$

- يبين الشكل التالي قيم تابع الترابط الذاتي r_k محسوب لقيم مختلفة للخطوة k:



b) متوسط الحزم batch mean

- يعتمد متوسط الحزم على تقسيم المخرجات x_1، x_2، x_3،....x_n إلى مجموعات بحيث تحتوي كل مجموعة على عدد متساوي من المخرجات عددها b مثلاً. أي أن المجموعة الأولى سوف تحتوي على المخرجات x_1، x_2، x_3،....x_b والمجموعة الثانية x_{b+1}، x_{b+2}، x_{b+3}،....x_{2b}، وهكذا. [٩ إيضاح]
- يتم حساب المتوسط للمجموعة الأولى $\overline{X}_1$، والثانية $\overline{X}_2$ ، وللمجموعة k بـ $\overline{X}_k$ فانه يمكن البرهان على أن المتوسطات غير مترابطة، ويعطى المتوسط الحزم بالعلاقة:

$$\overline{\overline{X}} = \frac{1}{k}\sum_{i=1}^{k}\overline{X}_i$$

- ويحسب الانحراف المعياري بالعلاقة:

$$s_d^2 = \frac{1}{k}\left(\sum_{i=1}^{k}(\overline{X}_i - \overline{\overline{X}})^2\right)$$

- من أجل القيم الكبيرة لـ k (مثلاً $k>30$) نحصل على مجال الثقة بالعلاقة:

$$\left(\overline{X} - 1.96\frac{s_d}{\sqrt{k}}, \overline{X} + 1.96\frac{s_d}{\sqrt{k}}\right)$$

[10 إيضاح]

c) طريقة التكرار replication

- يمكن حساب مجال الثقة باستخدام طريقة أخرى عن طريق تكرار المحاكاة عدة مرات ومن ثم حساب متوسط النتائج.
- لنفترض أننا ننفذ المحاكاة عدد n مرة:

التكرار -١: x_{11}، x_{12}، x_{13}،....x_{1m}

التكرار -٢: x_{21}، x_{22}، x_{23}،....x_{2m}

.

.

.

التكرار –n: x_{n1}، x_{n2}، x_{n3}،....x_{nm}

- يتم حساب المتوسط لكل تكرار باستخدام العلاقة:

$$\bar{X}_i = \frac{1}{m}\sum_{j=1}^{n} x_{ij}$$

- يتم حساب المتوسط للتكرارات على اعتبار أن كل تكرار مستقل بذاته:

$$\bar{\bar{X}} = \frac{1}{n}\sum_{i=1}^{n} \bar{X}_i$$

- ويحسب الانحراف المعياري بالعلاقة:

$$s_d^2 = \frac{1}{n}\left(\sum_{i=1}^{n} (\bar{X}_i - \bar{X})^2\right)$$

المشكلة تكمن في حساب طول التجربة أي قيمة m، و فترة العابرة لكل تجربة

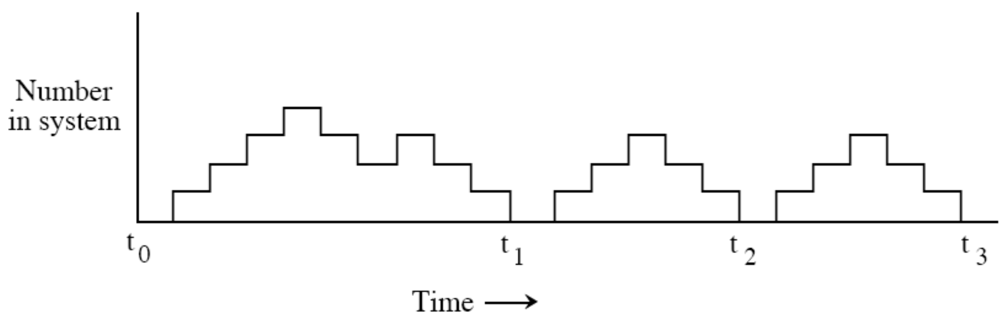
- يتم حساب التكرار بإتباع إحدى طريقتين:

✓ الأولى: بتنفيذ المحاكاة بقيم ابتدائية مختلفة لكل مرة تنفيذ والانتظار مدة كافية عند التنفيذ حتى يستقر النظام ومن ثم تحصيل النتائج.

✓ الثانية: بتنفيذ المحاكاة لمرة واحدة والبدء بتحصيل النتائج عند الوصول للحالة المستقرة وذلك لفترة محددة ومن ثم تحصيل النتائج مرة ثانية خلال نفس الفترة الزمنية الثانية وذلك دون التوقف عن العمل.

d) طريقة التجديد regenerative

- لاحظنا سابقاً أن طريقتي متوسط الحزم و التكرار تعتمدان فكرة توليد مخرجات مستقلة ومن ثم حساب متوسط هذه المخرجات المستقلة وذلك بتنفيذ المحاكاة مرات عدة، بينما تعتمد طريقة التجديد على توليد مخرجات مستقلة عن طريق تنفيذ المحاكاة لمرة واحدة.
- لنفترض أن t_0، t_1، t_2، ... هي الأزمنة التي يصبح بها النظام فارغاً، [11 إيضاح] ، حيث يرمز الزمن t_0 زمن دخول الزبون الأول لأول مرة في النظام، و الزمن t_1 زمن خروج الزبائن جميعها من الصف فيصبح المخدم فارغ لأول مرة.
- تسمى هذه النقاط نقاط التجديد regenerative points، وتسمى المسافة التي يقطعها المحاكي قبل أن يعود للحالة الفارغة بدورة التجديد regenerative cycle، كما هو مبين في الشكل التالي:



- لنفترض المتحول العشوائي للمخرجات X خلال المجال x_{i1}، x_{i2}، x_{i3}،....x_{in} وخلال زمنين الفارغين t_i و t_{i+1}، ولنفترض أن n_i، عدد المخرجات خلال لمجال المذكور. يمكن البرهان أن هذه المخرجات مستقلة عن المخرجات في الدورات المتجاورة.
- ليكن Z_i مجموع المخرجات خلال الدور i والذي يعطى بالعلاقة:

$$Z_i = \sum_{i=1}^{n_i} x_{ij}$$

- إذا يمكن حساب المتوسط المتوقع باستخدام العلاقة التالية:

$$Z = \frac{\frac{1}{M}\sum_{i=1}^{M} Z_i}{\frac{1}{M}\sum_{i=1}^{M} n_i}$$

حيث M عدد مرات دورات في المحاكاة.

2.4. التقييمات الأخرى للمتحول العشوائي

- حتى الآن تم حساب مجال الثقة للمتحول العشوائي لمخرجات النظام، في هذه الفقرة سوف نستعرض حساب التقييمات الأخرى والمفيدة أيضاً في تحليل نظام المحاكاة وهي:

a) احتمال وجود المتحول العشوائي ضمن مجال ثابت Probability that the random variable lie in fixed interval.

b) النسبة المئوية للمتحول العشوائي Percentile of random variable.

c) تباين المتحول العشوائي Variance of the random variable.

a) احتمال وجود المتحول العشوائي ضمن مجال ثابت Probability that the random variable lie in fixed interval.

- لتقييم ما إذا كان المتحول العشوائي يقع ضمن مجال ثابت نستخدم الطريقة نفسها المستخدمة لحساب المتوسط.
- ليكن لدينا I المجال المطلوب، وليكن X المتحول العشوائي إذا نريد أن نحقق:

$$p = \Pr(X \in I)$$

- نقوم أولاً بتنفيذ المحاكاة M مرة، بحيث يتم تحصيل N عينة من كل تنفيذ للمتحول العشوائي X، وليكن v_i عدد المرات التي يكون فيها المتحول العشوائي X يقع ضمن المجال I، إذاً فأن $p_i = v_i / N$ هو توقع المتحول العشوائي، إذا:

$$p = \frac{1}{M} \sum_{i=1}^{M} p_i$$

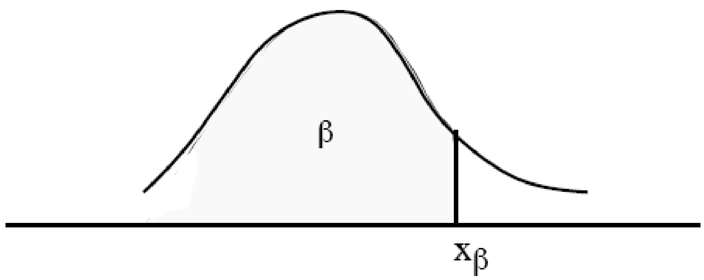
- ويحسب الانحراف المعياري بالعلاقة:

$$s_d^2 = \frac{1}{M-1}\left(\sum_{i=1}^{M} (p_i - \overline{p})^2 \right)$$

حيث نلاحظ أن حساب p تتطلب M تكرار للمحاكاة.

b) النسبة المئوية للمتحول العشوائي Percentile of random variable

- إن حساب النسب المئوية للمحول العشوائي مهم جداً في تحليل النظام، فمثلاً من غير المهم معرفة متوسط زمن الاستجابة لنظام ما، وإنما من المهم زمن الاستجابة نفسه (أي تخديم أكبر عدد ممكن من الآلات في أقل وقت ممكن). [12 إيضاح] .
- ليكن لدينا المتحول العشوائي X بتابع ذو كثافة احتمالي $f(x)$، تعرف النسبةالمئوية $100\beta^{th}$ بأنها أقل قيمة للمتحول العشوائي x_β بحيث $f(x_\beta)<\beta$ وبحيث أن المساحة المحصورة بين $-\infty$ و x_β وتحت المنحني $f(x)$ أقل أو تساوي القيمة β، كما هو مبين بالشكل:



- لنفترض انه قمنا بتكرار تنفيذ المحاكاة في كل مرة نتجت لدينا مخرجات مستقلة x_{i1}، x_{i2}، x_{i3}،....x_{iN} عددها N. لنقم بترتيب هذه المخرجات بحيث يكون $x_{ij}<x_{ij+1}$ عندها يتم حساب النسبةالمئوية $100\beta^{th}$ للمتحول العشوائي x_β عن طريق حساب المتوسط باستخدام العلاقة:

$$x_\beta = \frac{1}{M}\sum_{i=1}^{M} x_\beta^{(i)}$$

- إن تقييم النسبة المئوية يحتاج إلى تنفيذ المحاكاة بعدد مرات كثيرة وذلك للحصول على قيمة غير منحازة، إذا لحساب النسبة المئوية يجب:
  - ✓ تسجيل جميع المخرجات من بدء المحاكاة حتى نهاية التنفيذ.
  - ✓ ترتيب جميع المخرجات تصاعدياً.

[13 إيضاح]

c) تباين المتحول العشوائي Variance of the random variable

- لنفترض اننا قمنا بتنفيذ المحاكاة M مرة وفي كل مرة نتجت لدينا مخرجات مستقلة x_{i1}، x_{i2}، x_{i3}،....x_{iN} عددها N. يمكن حساب المتوسط في كل تنفيذ:

$$\overline{\mu}_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij}$$

- والمتوسط الكلي:

$$\overline{\overline{\mu}} = \frac{1}{M}\sum_{i=1}^{M} \overline{\mu}_i$$

- أي أن الانحراف المعياري لكل تنفيذ يساوي:

$$s_{d(i)}^2 = \frac{1}{N}\left(\sum_{j=1}^{N}(x_{ij} - \overline{\overline{\mu}})^2\right)$$

$$s_{d(i)}^2 = \frac{1}{N}\left(\sum_{j=1}^{N}(x^2{}_{ij} - 2\overline{\overline{\mu}}x_{ij} + \overline{\overline{\mu}}^2)\right) = \frac{1}{N}\left(\sum_{j=1}^{N} x^2{}_{ij} - 2\overline{\mu}_i\overline{\overline{\mu}} + \overline{\overline{\mu}}^2)\right)$$

- إذا فان التباين الكلي هو متوسط التباين لمجموع التنفيذ أي:

$$\sigma^2 = \frac{1}{M}\left(\sum_{i=1}^{M} s_{d(i)}^2\right)$$

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{N}\sum_{j=1}^{N} x^2{}_{ij}\right) - \frac{1}{M}\frac{2}{N}\sum_{i=1}^{M}\mu_i\overline{\mu} + \frac{2}{M}\overline{\mu}^2$$

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{N}\sum_{j=1}^{N} x^2{}_{ij}\right) - \overline{\overline{\mu}}^2$$

- ويمكن حساب التباين بتنفيذ المحاكاة لمرة واحدة، حيث يتم اعتبار في دورات أن المخرجات مستقلة ويتم من خلالها حساب قيم المتوسط.

5. طرق تقييم الحالات العابرة Estimation techniques for transient state

- تعتمد الحالة العابرة لنظام المحاكاة على الشروط الابتدائية، فللحصول على متحول عشوائي للحالات العابرة يجب تنفيذ المحاكاة مرات عدة، مع العودة للحالة الابتدائية في كل مرة وتنفيذ المحاكاة بنفس الشروط الابتدائية ومن ثم تسجيل المخرجات قبل أن يستقر النظام.
- لتحقيق ذلك يجب أن يكون مصمم النظام على دراية كافية بطول الفترة العابرة، بالإضافة لذلك يجب أن يتم استخدام مولد عشوائية ذو استقلالية تامة عن بقية مرات التنفيذ الأخرى.

6 التجارب الأولية Pilot experiments

27 حتى الآن بينا استعرضنا طرق عدة لحساب مجال الثقة، حيث لاحظنا أن مجال الثقة يتناسب عكساً مع عدد مرات تكرار التنفيذ N، فكلما كثرت مرات التنفيذ كلما صغر مجال الثقة المطلوب.

28 فوجد أن النسبة تساوي $1/\sqrt{N}$ فلتخفيض مجال الثقة إلى النصف تحتاج إلى زيادة عدد مرات التكرار أربع أضعاف $1/\sqrt{4N} = 0.5(1/\sqrt{N})$ و كلما صغر مجال الثقة كلما زادت الدقة للنظام.

29. في معظم الأحيان لا يملك مصمم النظام أية فكرة على عدد مرات التكرار N الواجب تطبيقه وذلك للحصول على قيمة دقة مطلوبة، لذلك يتم في كثير من الأحيان إجراء تجارب أولية لإعطاء مصمم النظام القيمة المتوقعة لـ N عن طريق حساب مجال الثقة للتجربة.

ملحقات

[1 مثال]

مثلاً معرفة عدد مرات تكرار تعطل آلة ما في مصنع، أو زمن إصلاح الآلة ما عند وجود عامل معين ... .

[2 مثال]

مثلاً يمكن حساب زمن تعطل آلة بأنه مجموع الأوقات التي بقية فيه الآلة في فترة انتظار الإصلاح بالإضافة إلى زمن الإصلاح.

[1 إيضاح]

في مثال إصلاح الآلات في مصنع يمكن حساب زمن تعطل آلة باستخدام المعلومات التالية:

- زمن وصول الآلة إلى صف الانتظار (وقت التعطل).
- زمن خروج الآلة من الإصلاح (وقت الإصلاح).

يمكن حفظ هذه المعلومات في مصفوفة، وعند انتهاء المحاكاة تستخدم هذه المصفوفة في حساب زمن توقف الآلات، وحساب معلومات إحصائية أخرى مثل متوسط زمن التوقف mean و الانحراف المعياري standard deviation و نسب التعطل.

[2 إيضاح]

يمكن الاكتفاء باستخدام مصفوفة أحادية حيث يتم تخزين زمن الوصول عند وصول الرزمة إلى العقدة، وعند مغادرة الرزمة للعقدة يتم طرح زمن الخروج من الزمن المخزن أصلاً في المصفوفة والذي يعبر عن زمن الوصول وتخزين الناتج في المصفوفة لنحصل على زمن بقاء الرزمة في المصفوفة.

[3 إيضاح]

طبعاً يتغير أداء المحاكاة في كثير من الأحيان مع تغير الشروط الابتدائية.

[4 إيضاح]

إذا تمت المحاكاة خلال فترة طويلة فان أداء النظام يكون مستقلا عن الشرط الابتدائية، ويكون التأثر في حال أجريت المحاكاة خلال فترة زمنية صغيرة محددة T.

[5 إيضاح]

إذا لمعرفة الشروط الأولية بدقة يجب أن يكون لدى مصمم النظام خبرة دقيقة بالنظام من حيث مدخلاته ومخرجاته وتوقع أداء عمله.

[1 مثال]

لقد بينا في المثال إصلاح الآلات في المصنع (راجع الفصل الأول) انه يوجد حدثين فقط، الأول دخول الآلة إلى صف الإصلاح (أي تعطل الآلة) والثاني خروج الآلة من الصف (أي إصلاحها). حيث تدعى هذه الأحداث بالأحداث الأولية Primary events.

[2 مثال]

إن تكرار حدوث أعطال في الآلات ودخول الآلات إلى صف الانتظار قد يؤدي إلى حدوث حالة جديدة لا تخرج الآلات من صف الانتظار (إي لا تصلح) وذلك عندما يكون عامل الإصلاح بحاجة إلى وقت للراحة idle حتى ولديه عدد كبير من الآلات بحاجة إلى إصلاح.

[5 إيضاح]

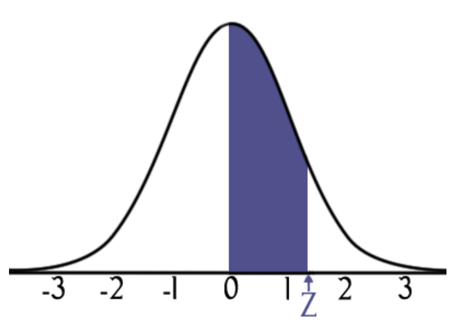
سوف نشرح قائمة الأحداث المستقبلية في الفقرة التالية.

[6 إيضاح]

يعكس مجال الثقة قيمة الخطأ الممكن حدوثه عند حساب المتوسط. أي انه عند تنفيذ التجربة لمئة مرة فأن 95% من هذه التجارب سوف تكون ضمن المجال الصحيح.

[7 إيضاح]

يمكن حساب أية نقطة من نقاط المنحني باستخدام جدول التوزع القياسي المبين:



	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.0000	0.0040	0.0080	0.0120	0.0160	0.0199	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.0910	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.1480	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.1700	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.1950	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.2190	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2517	0.2549
0.7	0.2580	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.2910	0.2939	0.2967	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.3340	0.3365	0.3389
1.0	0.3413	0.3438	0.3461	0.3485	0.3508	0.3531	0.3554	0.3577	0.3599	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.3770	0.3790	0.3810	0.3830
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.3980	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.4370	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.4750	0.4756	0.4761	0.4767
2.0	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.4830	0.4834	0.4838	0.4842	0.4846	0.4850	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.4890
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.4920	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.4940	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.4960	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.4970	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.4980	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3.0	0.4987	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.4990	0.4990

[8 إيضاح]

ويتم حساب المعادلة السابقة باستخدام العلاقة التالية:

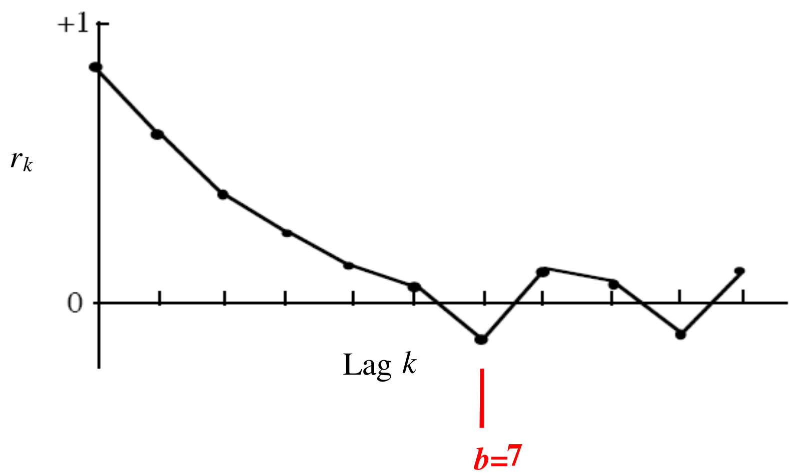
$$c_{xy} = \sum_{n=0}^{N-1} (x_i - \mu_x)(y_i - \mu_y)$$

[9 إيضاح]

ويتم اختيار المجموعات بحيث تكون المخرجات فيما بينها مترابطة ضمن المجموعة الواحدة

[10 إيضاح]

يجب اختيار عدد العينات ضمن كل حزمة b اكبر ما يمكن بحيث لا تكون قيم المخرجات ضمن الحزمة غير مترابطة، ويتم اختيار قيمة b عن طريق رسم المنحني البياني لقيم تابع الترابط الذاتي ومن ثم اختيار القيم التي تكون فيه b اصغر ما يمكن:



[11 إيضاح]

وهي الحالات التي يخرج بها الزبون من الصف فيصبح مخدم النظام فارغاً بحيث لا يوجد أية زبون في الصف.

[12 إيضاح]

مثلاً يتم حساب نسبة الاستجابة يجب أن لا تقل في نظام ما عن 95%.

[13 إيضاح]

يمكن الحصول بشكل مباشر على قيمة النسبة المئوية عن طريق رسم الهيستوغرام للمخرجات ومن ثم حساب مجال الثقة المطلوب ومنه استنتاج القيمة المطلوبة.

مسائل

-13 قم بتعديل برنامج المحاكاة الذي قمت بتصميمه سابقاً حول محاكاة مصنع، بحيث تحقق مايلي:

- a- قم بتحصيل المعلومات بالزمن الذي تقضيه الآلة في فترة الإصلاح، وفي فترة العمل.
- b- قم بإضافة برنامج جزئي يقوم بحساب متوسط و الانحراف المعياري.
- c- قم بتنفيذ المحاكاة لـ ٥٠ مرة واعتبر أنها فترة عابرة، واحسب المتوسط والانحراف المعياري.
- d- تابع تنفيذ المحاكاة لـ 5٥0 مرة واعتبر أنها فترة المستقرة، واحسب المتوسط والانحراف المعياري.
- e- قارن بين النتائج السابقة و سجل ملاحظاتك.
- f- استخدم طريقة حساب متوسط الحزم لحساب المتوسط، وقارن النتائج التي حصلت عليها مع النتائج السابقة.
- g- احسب مجال الثقة لـ 95% من تعطل الآلة.

لغات المحاكاة

Simulation Languages

1. مقدمة

- بالعودة إلى الفصول السابقة نجد أنه عند تصميم نظام محاكاة يتطلب من لغة البرمجة أن تحقق مجموعة من البرامج المشتركة يمكن أن نحددها بما يلي:
  - برامج جزئية توليد الأعداد العشوائية الزائفة بموزعة بتوزيع منتظم.
  - برامج جزئية توليد الأعداد العشوائية الغير محدد موزعة بتوابع توزيع متعددة.
  - برامج جزئية لتوليد والتحكم بساعة المحاكي.
  - برامج جزئية لتحديد الحدث التالي أو الجديد من قائمة الأحداث المستقبلية.
  - برامج جزئية للتحكم بقائمة الأحداث المستقبلية.
  - برامج جزئية لتجميع وإظهار النتائج.
- وكما سنرى لاحقاً نحتاج إلى برامج لدراسة النتائج وتحليلها وبرامج لكشف الأخطاء.
- يمكن باستخدام لغات البرمجة عالية المستوى C أو C++ أو J++ تحقيق هذه البرامج بسهولة، ولكن يمكن استخدام حزم برمجية خاصة يمكن من خلالها تصميم أية نظام محاكاة بكفاءة وسرعة عالية وبدون الدخول بالتفاصيل البرمجية الدقيقة.
- سوف نستعرض في هذا الفصل فكرة عن عدد أشهر هذه الحزم البرمجية من خصائص ومزايا ومساؤى هذه الحزم بدون الدخول بتفاصيل كيفية التعامل معها واستخدامها.

2. مقارنة بين الحزم البرمجية ولغات البرمجة

- إن أصعب قرار يمكن أن يأخذه مصمم النظام هو الاختيار بين استخدام لغات برمجة عالية المستوى وبين استخدام حزم برمجية جاهزة لتصميم نظام المحاكاة المطلوب. وأيضاً بين اختيار الحزمة الأنسب لنظامه المطلوب تحقيقه.
- وتتميز الحزم البرمجية الجاهزة والخاصة بتصميم نظام محاكاة بالميزات التالية:

✓ تقدم الحزم البرمجية برامج جزئية جاهزة والتي يتطلبها نظام محاكاة [1 إيضاح] ، مما يقلل بشكل كبير الوقت والجهد اللازمين لتحقيقها.

✓ تهئ للمبرمجين إطار عمل وقالب عمل واحد وجاهز، وهذا الإطار أو القالب مناسب لنظم المحاكاة، بينما لا تقدم لغات البرمجة عالية المستوى هذا الإطار.

✓ تعطي الحزم البرمجية نظام محاكاة قابلة للتعديل و التغيير بسهولة.

✓ تقدم الحزم البرمجية وسائل برمجية مهمتها كشف الأخطاء أوتوماتيكيا.

- من جهة أخرى بعض نظم المحاكاة تستخدم لغات البرمجية عالية المستوى لعددت أسباب منها:
  - خبرة المبرمجين الكبيرة بلغات البرمجة عالية المستوى، ولكن عدم معرفتهم بالحزم البرمجية.
  - إن النظم البرمجية المصممة بلغات برمجة عالية المستوى تتميز بزمن تنفيذ صغير مقارنتاً بزمن التنفيذ الذي تأخذه برامج المحاكاة المصممة بالحزم البرمجية الجاهزة.
  - تهئ لغات البرمجة عالية المستوى مرونة اكبر بالتصميم والتحقيق من الحزم البرمجية المحدودة ضمن قوالب خاصة.
  - إن لغات البرمجة C++ و J++ هي لغات غرضية التوجه وهذا غير متوفر في معظم الحزم البرمجية الجاهزة.
  - إن الكلفة الاقتصادية العالية للحزم الجاهزة تؤدي في كثير من الأحيان إلى استخدام لغات برمجة عالية المستوى.

3. تصنيف الحزم البرمجية

- في هذه الفقرة سوف نستعرض بالتفصيل أصناف الحزم البرمجية حيث سنبين ما يلي:
  - مقارنة بين الحزم البرمجية لأغراض عامة أو لتطبيقات خاصة.
  - العوامل المشتركة بين الحزم البرمجية.

1.3 مقارنة بين الحزم البرمجية لأغراض عامة أو لتطبيقات خاصة

- تصنف الحزم البرمجية الخاصة بنظم المحاكاة إلى نوعين أساسيين: الأول لغات المحاكاة Simulation languages، والثاني الحزم البرمجية الخاصة بتطبيقات خاصة application-oriented simulation.
  - لغات البرمجة تتميز بالمرونة العالية في تحقيق النظام المطلوب ولكنها بنفس الوقت تحتاج لمطلوب ولكنها بنفس الوقت تحتاج إلى خبرة كبيرة في استخدام اللغة.
  - الحزم البرمجية الخاصة تتميز بأنها سهلة الاستخدام ويمكن تحقيق المحاكاة بسهولة، ولكنها موجه لتطبق معين فقط.
- ولكن خلال السنوات الأخيرة قام مطورو لغات البرمجة بجعل برامجهم أكثر سهولة وذلك باستخدام الرسوم، واستخدام الماوس في إضافة وحذف أية عنصر من عناصر المحاكاة بسهولة وتغيير مواصفات أية عنصر. [2 إيضاح]

2.3 العوامل المشتركة بين الحزم البرمجية

- تحتوي الحزم البرمجية على مجموعة من العوامل المشتركة منها:

✓ الكينونات entities: وهي العناصر التي تسافر أو تتحرك في نظام المحاكاة ويمكن أن تدمر أو تختفي في النهاية، الطائرة في نظام محاكاة مطار مثلاً.

✓ الصفات attributes: وهي الصفات التي تتميز بها الكينونات، وهي المعلومات الخاصة التي تخزن في كل كينونة. رقم الرحلة نوع الطائرة..الخ في نظام محاكاة مطار.

✓ الموارد resources: وهي موارد نظام المحاكاة والتي تستخدمها الكينونات أثناء تحركها في النظام. البوابة في مطار أو مدرج الطيران في نظام محاكاة مطار.

✓ الصفوف queues: وهي أماكن انتظار الكينونات عندما تكون الموارد غير متوفرة، ويمكن أن يتم تخديم هذه الكينونات بطريقة FIFO أو بطريقة LIFO أو بالترتيب التصاعدي أو التنازلي. مثلاً انتظار الطائرات على مدرج المطار للإقلاع في نظام محاكي مطار.

- يبين الجدول التالي مجموعة أمثلة للعوامل المشتركة بين الحزم البرمجية:

الصفوف	الموارد	الصفات	الكينونة	طبيعة النظام
الصف او التخزين المؤقت،	الألات، العمال،..	رقم القطعة، تاريخ الطلب،..	قطعة	مصنع
عازل buffer	العقد، الروابط،..	جهة الرسالة، طول الرسالة،..	رسالة	اتصالات
صف queue	مدرج الطيران، البوابة،..	رقم الرحلة، وزن الرحلة،..	طائرة	مطار
صف queue	مكاتب الزبائن، الموظفين،..	اسم، رقم ضبط الشرطة، الكلفة،..	طلب زبون	شركة تأمين

4. مواصفات لغات البرمجة

- عند اختيار لغة برمجة لتصميم محاكاة لنظام ما، يجب أن تتمتع هذه اللغة بمجموعة مواصفات خاصة تمكنها من تحقيق نظام المحاكاة بشكل كامل. هذه المواصفات يمكن أن نلخصها بما يلي:
  A. العمومية.
  B. مراعاة المواصفات البرمجية و العتادية .
  C. قدرة بناء رسوم متحركة.
  D. الخاصية الإحصائية.
  E. خدمة الزبون الوثائق المساعدة.
  F. قدرة إصدار تقارير بالنتائج.
- سوف نستعرض هذه المواصفات أكثر تفصيلاً في الفقرات القادمة.

A- العمومية

- يقصد بالعمومية هو قدرة لغة البرمجة من تحقيق نظام ما مهما كانت درجة تعقيد هذا النظام، حيث تجهز لغة البرمجة بإمكانية إضافة برامج جزئية مكتوبة بلغات برمجة عالية المستوى مثل C مثلاً ليتمكن المصممين من تحقيق أية حدث غير متوفر في لغة البرمجة أصلا.
  - ✓ لتكون لغة البرمجة مرنة يجب أن تحقق الخصائص التالية:
  - ✓ يجب أن يتمكن المبرمج من تغير خصائص الكينونة بسهولة، ومن تغيير قيم المتحولات العامة أيضاً.
  - ✓ يجب أن تحتوي لغة على التعليمات الشرطية if-then-else.
  - ✓ القابلية على برمجة معادلات رياضية و توابع رياضية معقدة مثل اللغارتيمات والجذور.
  - ✓ القابلية على إنشاء شروط تحكم جديدة وتخزين الشروط القديمة في قواعد بيانات خاصة.
- من أهم المواصفات التي يجب أن تتوفر في لغة البرمجة هي سهولة في الاستخدام easy of uses و السهولة في التعلم easy of learning، فمعظم لغات البرمجة تحتوي على Graphical user interface لتسهيل التخاطب مع المبرمج.

B- مراعاة المواصفات البرمجية و العتادية

- عند اختيار لغة البرمجة ما هي مواصفات الجهاز الحاسوبي المناسب الممكن استخدامه وما هو نظام التشغيل المستخدم. فيجب مراعاة مالي:
  - ✓ نظام التشغيل المستخدم من Unix، Linux، أو windows.
  - ✓ الجهاز من نوع Apple أو من IBM Compatible.
  - ✓ حجم ذاكرة RAM المستخدمة.
  - ✓ حجم وقدرة كرت الإظهار المستخدم.
  - ✓ سرعة المعالج وقوته.

C- قدرة بناء رسوم متحركة

- إن قدرة نظام لغة البرمجة للمحاكي على بناء رسوم متحركة للنظام المراد برمجته هي من أهم الميزات المراد التي يجب أن تتوفر في لغة البرمجة.
- وتعني قدرة بناء رسوم على أن المحاكي يستخدم ايكونات تتغير مواقعها وألوانها ومواقعها على الشاشة مع الزمن عند تشغيل البرنامج وبالتناغم مع تغيرات متحولات المحاكي بحيث تعكس بشكل رسومي ما يجري حقيقتاً. (مثلاً استخدام أيكون يمثل رافعة في معمل تتحرك عند طلب مادة ما من المستودع).
- ويجب أن يضمن هذه اللغة على ما يلي:

✓ وسائل التخاطب مع المستخدم للتحكم والحصول على النتائج، وبحيث أن المستخدم غير خبيرة في التصميم والبرمجة.

✓ وسائل تصحيح وكشف الأخطاء.

✓ وسائل إظهار ما إذا كان المحاكي يعطي نتائج صحيحة أم لا.

✓ وسائل لتدريب المستخدمين.

✓ وسائل التخاطب بين المستخدمين.

D- الخاصية الإحصائية

- يجب ان يحتوي النظام على وسائل إحصائية تقوم بإصدار وتحليل النتائج وذلك للحصول على نتائج دقيقة وقرارات دقيقة لنظام المحاكاة.
- ليتمكن نظام المحاكي من مطابقة الواقع بشكل تقريبي يجب أن يحتوي على مولد للأعداد العشوائية ذات توزيع منتظم، بحيث تمر هذه الأعداد على الاختبارات المستخدم للتحقق من توزعها المنتظم واستقلاليتها.
- يجب أن يحتوي المحاكي على الوسائل الخاصة بتوليد أعداد عشوائية بتوزيعات مختلفة (أسي، غاما، طبيعي...) بناء على رغبة المبرمج.
- يجب أن يحتوي المحاكي على وسيلة لتشغيل المحاكاة عدة مرات منفصلة بحيث يضمن تشغيلاً بأعداد عشوائية مختلفة في كل تشغيل حتى عند استخدام نفس الشروط الابتدائية.

E- خدمة الزبون الوثائق المساعدة

- يجب أن تحتوي لغة البرمجة على وثائق مساعدة تمكن المبرمج من الرجوع إليها عند الطلب ،
- يجب أن تحتوي على أمثلة تفيد في التعرف على كيفية بناء نظام محاكي
- يجب أن يحتوي لغة البرمجة على وسيلة تقدم خدمة للزبون سواء عن طريق الهاتف أو عن طريق الانترنت online support.
- معظم لغات البرمجة تحتوي على demo disk يوزع مجانا ليسهل على متخذي القرار اختيار أفضل محاكي المناسب.

F- قدرة إصدار تقارير بالنتائج

- يجب على المحاكي أن يكون قادراً على إصدار تقارير قياسية متعارف عليها وذلك لقياس أداء النظام.
- يجب أن يكون قادراً على تعديل أو تصميم تقارير خاصة بالمستخدم.
- يجب أن يكون قادراً على رسم النتائج بيانيا، (تغيرات المتحولات مع الزمن، الهيستوغرام Histogram، الترابط بين متحولات النظام، ...).

5. الحزم البرمجية عامة التوجه General purpose Simulation packages

- سوف نستعرض في هذه الفقرة فكرة عامة ومختصرة عن الحزم البرمجية عامة التوجه والتي تستخدم لبناء نظام محاكاة. سوف نقدم مثالين فقط لأشهر أنواع هذه الحزم والتي تمتع بالموصفات المبينة سابقا:
  - ✓ الحزمة الأولى تسمى Arena من شركة Rokwell Automation [3 إيضاح]
  - ✓ الحزمة الثانية تسمى Extend من شركة Imagine [4 إيضاح]

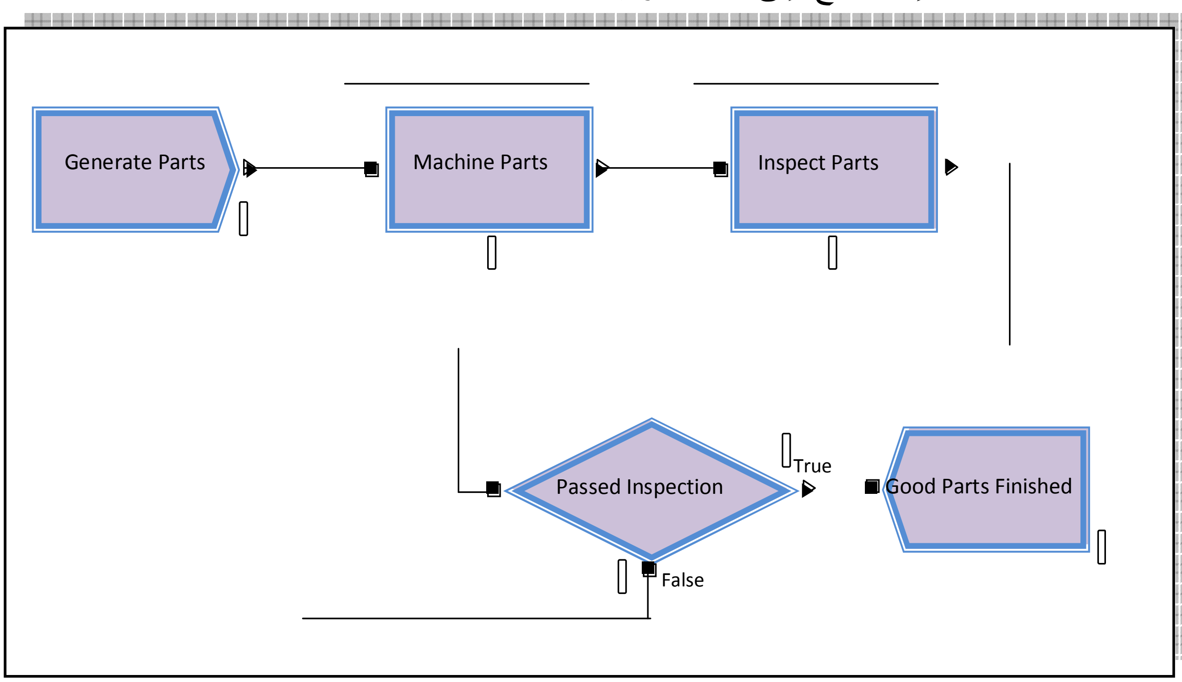
- سوف نقوم خلال الفقرات القادمة بإعطاء فكرة بسيطة حول كيفية بناء نظام محاكة بسيط باستخدام الحزمتين السابقتين.

1.5 حزمة Arena

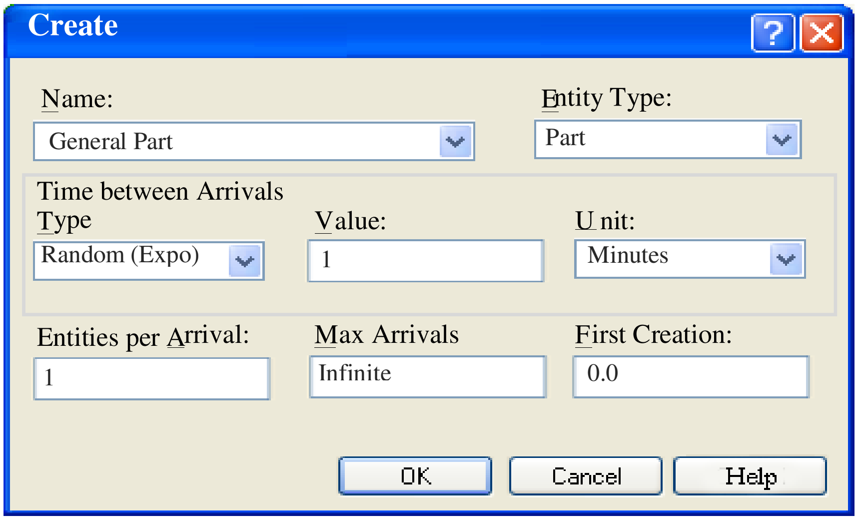
- تعتبر حزمة Arena من البرمجيات المستخدمة للمحاكاة عامة التوجه، وهو مستخدم لتصميم محاكاة لعدد متنوع من التطبيقات منها:
  - ✓ المصانع.
  - ✓ خطوط الإنتاج.
  - ✓ في القطاع العسكري.
  - ✓ في القطاع الصحي.
  - ✓ في مراكز خدمة الزبائن.
- ويتوفر Arena بعدد من الإصدار ويتراوح من حزمة بسيطة أساسية Basic لتصميم المحاكاة البسيطة إلى حزمة محترفة لتصميم نظم محاكاة معقدة.
- تتألف حزمة Arena من مجموعة من المركبات modules تغطي عدد كبير من القوالب templates والتي هي عبارة عن برامج جاهزة منها:
  - ✓ العمليات المنطقية logic والروابط
  - ✓ واجهات التخاطب مع المستخدم
  - ✓ الرسومات المساعدة.
  - ✓ معالجة دخول و خروج الكينونات
  - ✓ معالجة معطيات مدخلة من برامج اخرى مثل Excel، Access، و SQL.

- وتتم البدء بالمحاكاة عن طريق جر الموديولات إلى النافذة الرئيسية ومن ثم القيام بوصل الروابط بينها لإنشاء المخطط التدفقي الخاص بالمحاكاة.
- و يتوفر في Arena عدد كبير من مولد الأعداد العشوائية المبينة في الفصول السابقة، وتحتوي أيضاُ على أكثر من 12 تابع توزيع كثافة احتمالي.
- كما يتوفر في الحزمة طرق لتكرار المحاكاة أتوماتيكياً وحساب مجال ثقة للمتوسط وفق الطرق المبينة في الفصل السابق.من ثم برمجة كل موديول عن طريق إدخال المعلومات و الأوامر الخاصة بكل موديول.

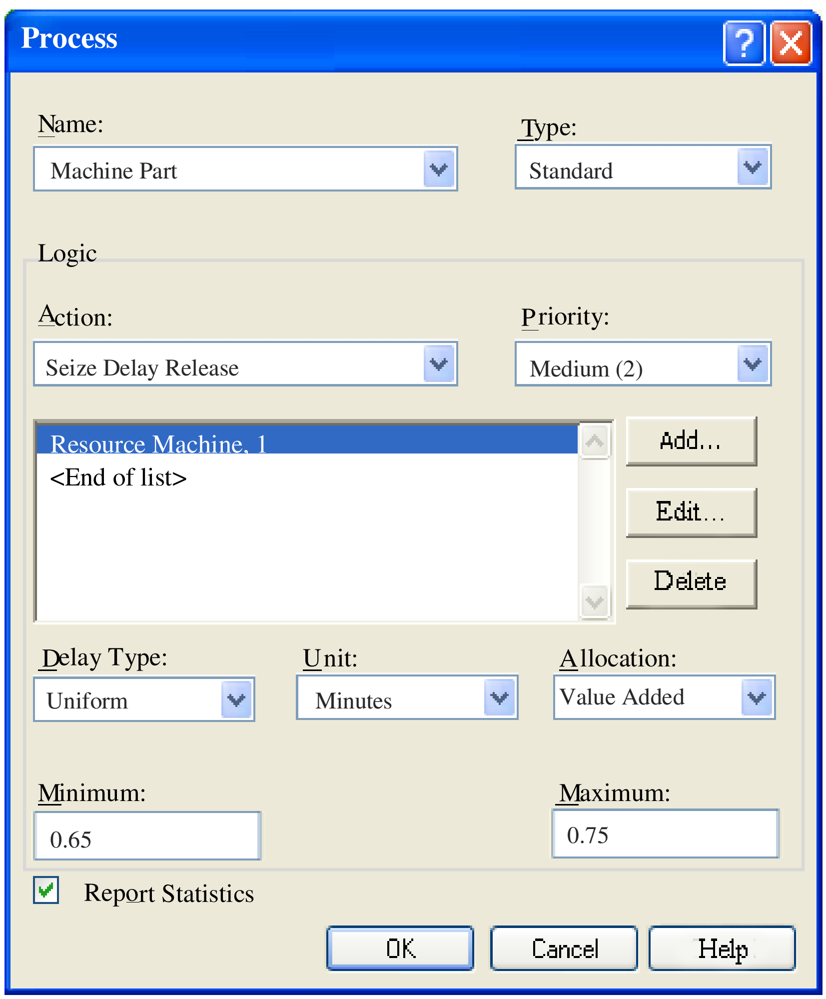
- سوف نشرح بشكل مختصر بناء نظام محاكي لخط إنتاج في مصنع، ومن ثم يقوم عامل باختبار صلاحية قطعة المنتجة، حيث نفترض أن خط الإنتاج لا يتوقف ابدا وان الآلات لا تتعطل.
- يبين الشكل التالي بناء المخطط التدفقي لمحاكي لخط إنتاج في مصنع مبني بحزمة Arena، حيث نلاحظ إننا نحتاج إلى خمسة موديلات.



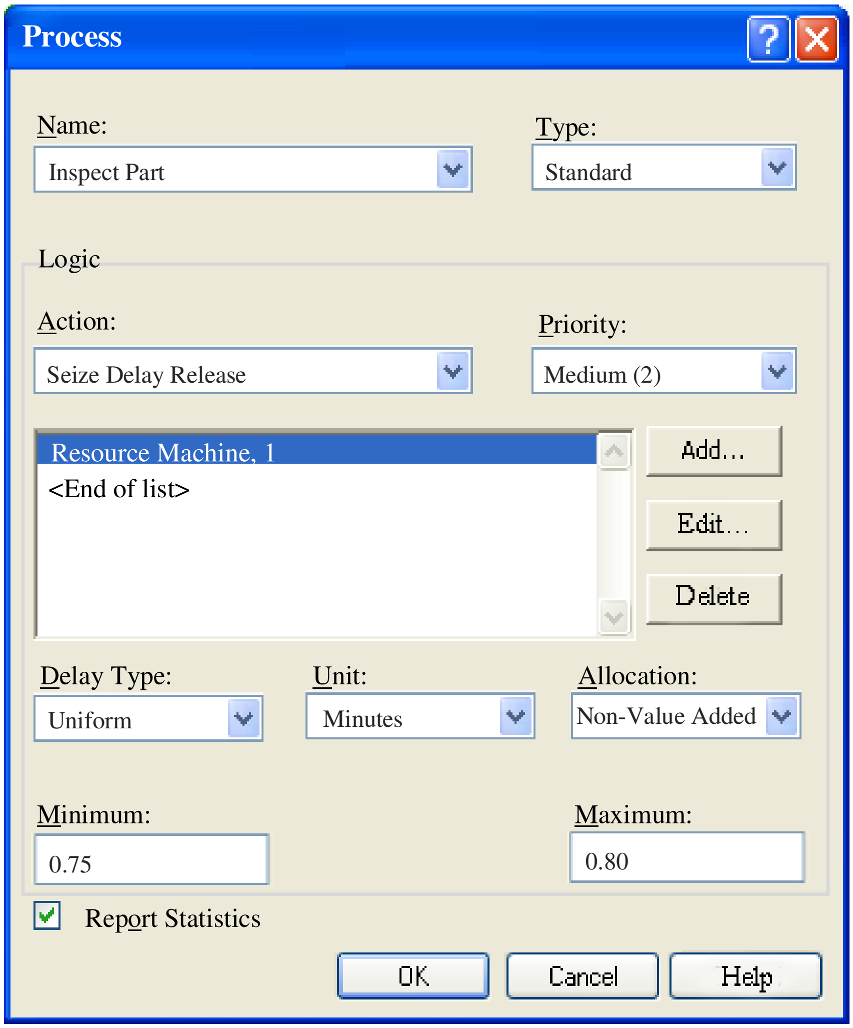
- يستخدم الموديول Generate Parts لتوليد الأجزاء إلى خط الإنتاج، ويكون معدل توليد الأجزاء عشوائياً بتابع توزيع كثافة احتمالي أسي و بمتوسط يساوي 1 دقيقة، يبين الشكل التالي النافذة الخاصة بخصائص هذا الموديول.



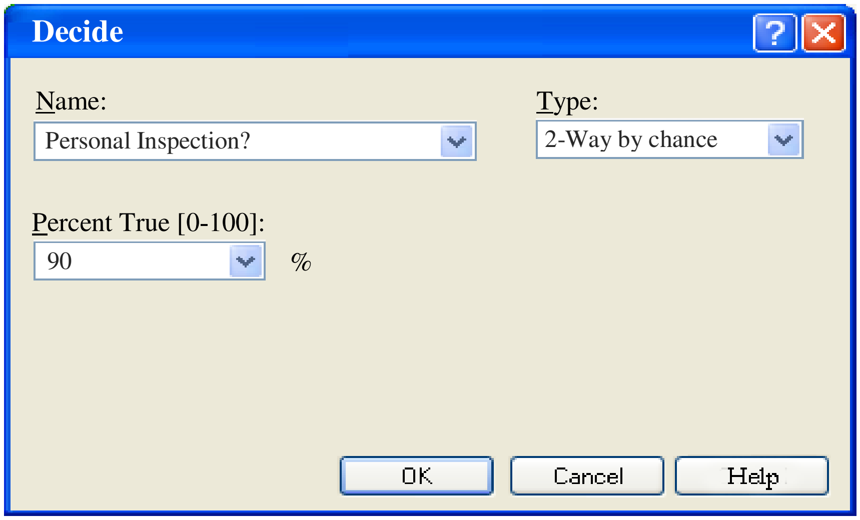
- يرتبط الموديول Create Module بالموديول الثاني Machine Parte والمسؤول عن تركيب أو التعامل مع الأجزاء Process، ولهذا الموديول له تابع توزع احتمالي منتظم بين القيم 0.65 و 0.75 دقيقة، يبين الشكل التالي النافذة الخاصة بخصائص هذا الموديول.



- الموديول التالي Inspector Module وهو المسؤول عن اختبار المنتج بعد التصنيع في الموديول الثاني Machine Parte، و يتبع هذا الموديول تابع توزع احتمالي منتظم بين القيم 0.75 و 0.8 دقيقة، يبين الشكل التالي النافذة الخاصة بخصائص هذا الموديول.



- الموديول التالي هو Inspector وهو المسؤول عن وضع القرار فيما إذا كان المنتج صالح أم غير صالح، حيث يتم أختيار أن 90% من المنتجات صالحة، يبين الشكل التالي النافذة الخاصة بخصائص هذا الموديول.



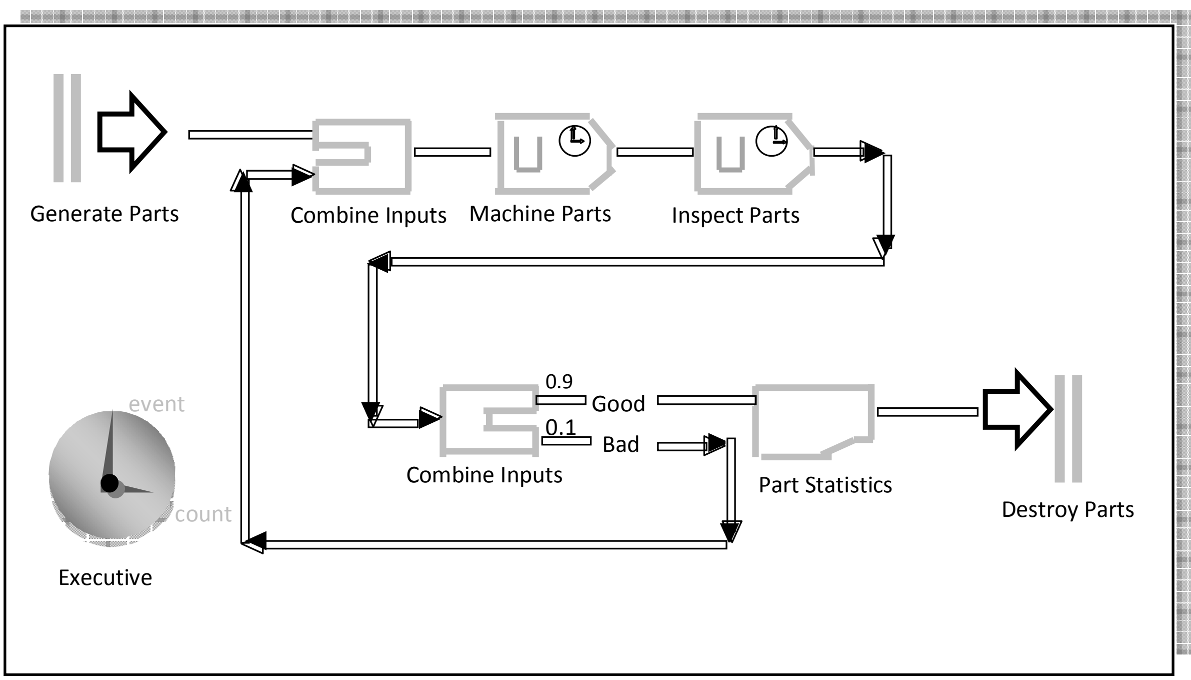
- ويلي هذا الموديول موديول Good Part Finished والمسؤول عن تغليف المنتجات النهائية للشحن مثلاً، أم الأجزاء الغير صالحة فتعود إلى دورة التصنيع من جديد.

30. يتم بعد ذلك تنفيذ المحاكاة بمدة مثلاً 100000 دقيقة، يبين الجدول التالي النتائج التي حصلنا عليها.

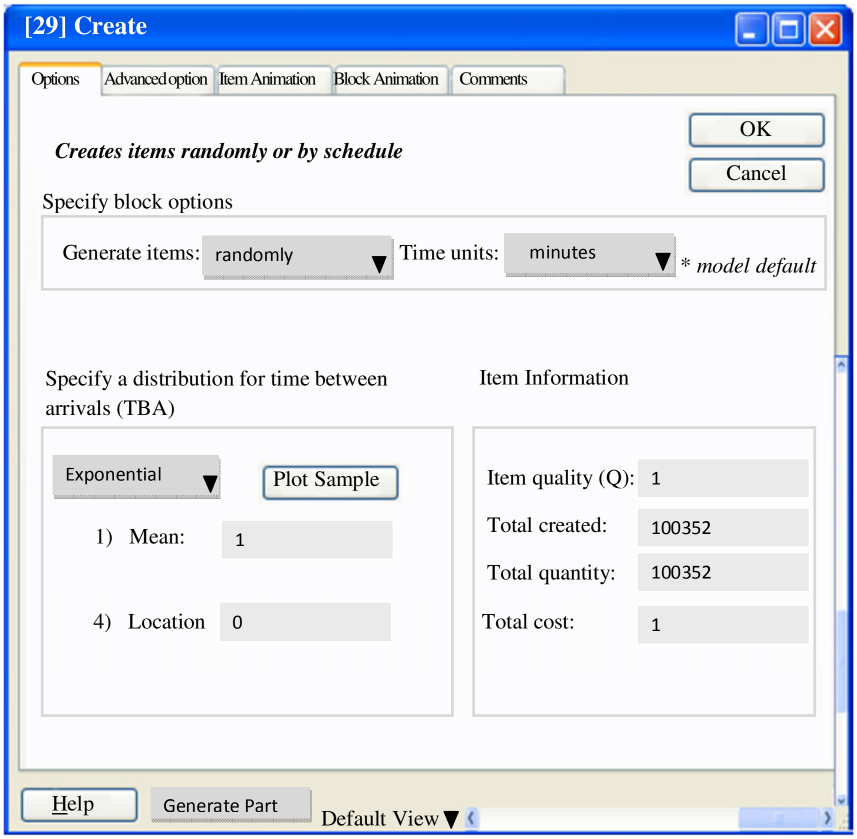
	Average	Half Width	Minimum Value	Maximum Value
Waiting Time	3.0280	0.247628038	0.00	94.4532
Transfer Time	0.00	0.000000000	0.00	0.00
Other Time	0.00	0.000000000	0.00	0.00
Total Time	4.6397	0.248893749	1.4001	101.70

2.5 حزمة Extendsim

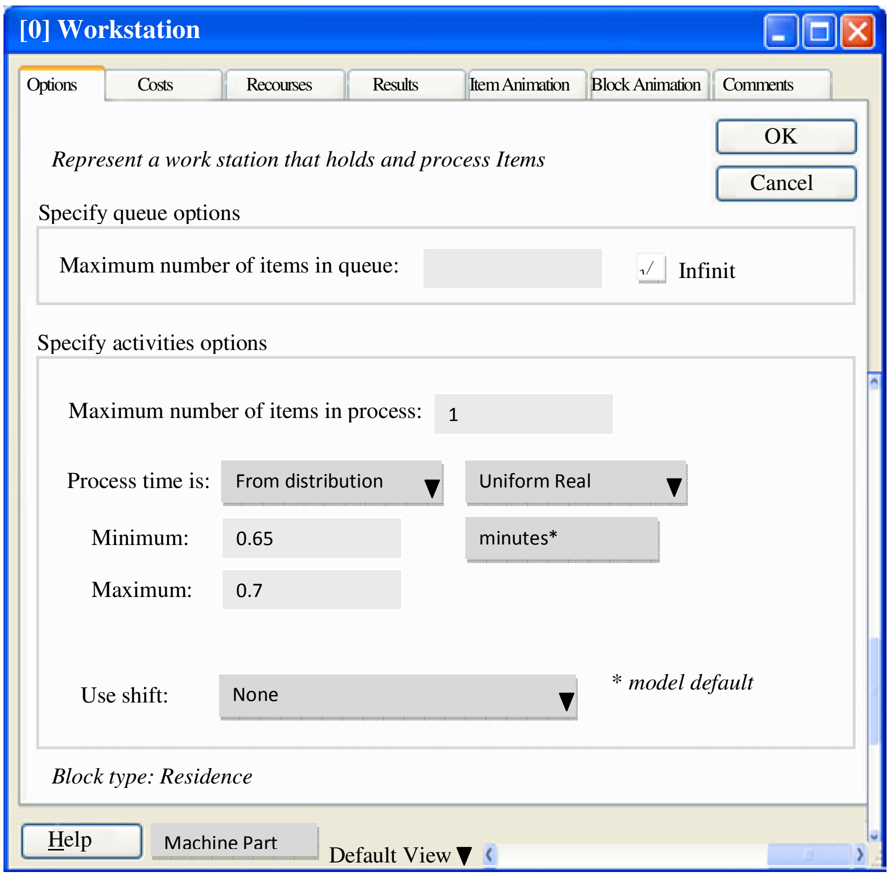
- تعتبر حزمة Extendsim من البرمجيات المستخدمة للمحاكاة عامة التوجه، وهو مستخدم لتصميم محاكاة لعدد كبير من التطبيقات بسبب الخاصية التي يتمتع بها من إضافة مكتبة خاصة بالمستخدم وذلك باستخدام لغة ++C أو لغة Visual Basic ويمكن للمستخدم تعديل البرنامج أو قراءته بمجرد الضغط على الموديول بمفتاح الفأرة الأيمن.
- يعتمد بناء محاكي على نظام هرمي تتوسع قاعدته بشكل كبير، بحيث أن كل مودل model يمكن توسيعه إلى مجموعة متكاملة من الموديلات models، وبحيث ان توسيع المودل يتم بنوافذ رسومية جديدة للتحكم.
- يتوفر في Extendsim أيضاً عدد كبير من مولد الأعداد العشوائية المبينة في الفصول السابقة، ويحتوي أيضاُ على أكثر من 34 تابع توزيع كثافة احتمالي.
- كما يتوفر في الحزمة طرق لتكرار المحاكاة أتوماتيكياً وحساب مجال ثقة للمتوسط وفق الطرق المبينة في الفصل السابق.من ثم برمجة كل موديول عن طريق إدخال المعلومات و الأوامر الخاصة بكل موديول.
- سوف نشرح بشكل مختصر بناء نظام محاكي لخط إنتاج في مصنع باستخدام Extendsim، ومن ثم يقوم عامل باختبار صلاحية قطعة المنتجة، حيث نفترض أن خط الإنتاج لا يتوقف ابدأ وان الآلات لا تتعطل
- يبين الشكل التالي بناء المخطط التدفقي لمحاكي لخط إنتاج في مصنع مبني بحزمة Extendsim:



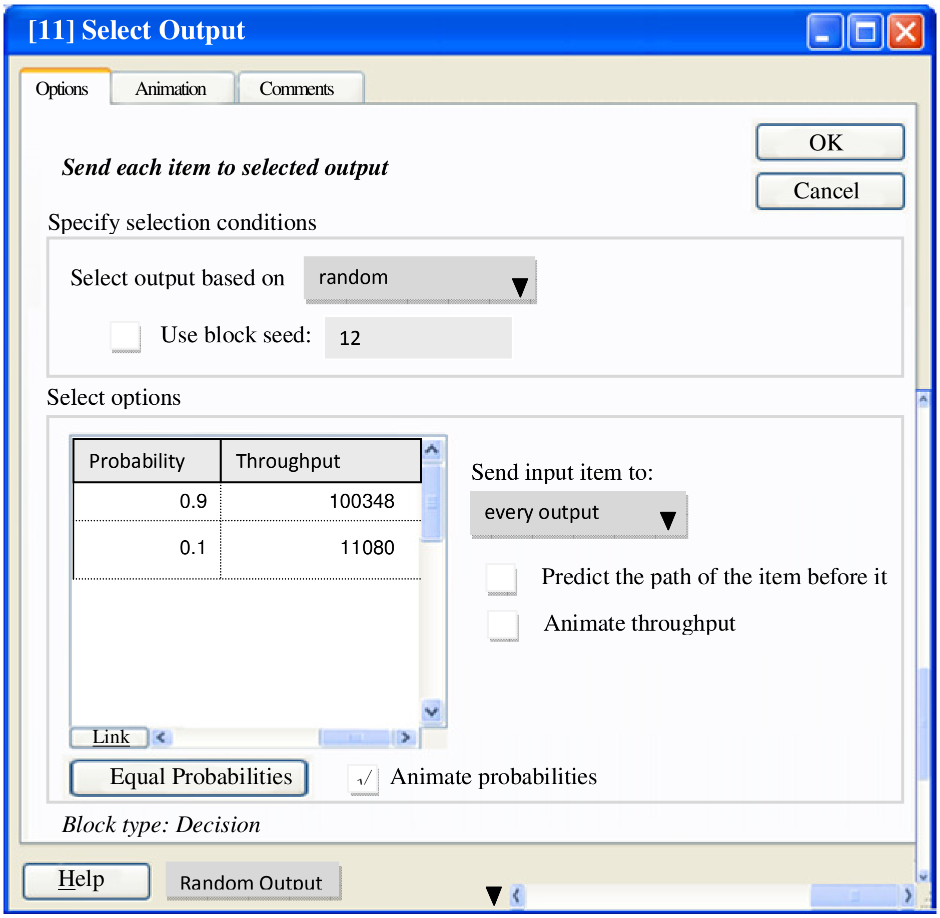
- تسمى الكتلة block الأولى بـ Create Parts وهي المسؤولة عن توليد الأجزاء إلى خط الإنتاج، ويكون معدل توليد الأجزاء عشوائياً بتابع توزيع كثافة احتمالي أسي وبمتوسط يساوي 1 دقيقة، يبين الشكل التالي النافذة الخاصة بخصائص هذه الكتلة.



- ترتبط الكتلة الأولى Create Block بالكتلة الثانية Combine Inputes ومهمتها دمج الأجزاء القادمة من الكتلة الأولى أي المولدة وبين الأجزاء التالفة، وتعطي في خرجها سلسلة واحدة.
- تربط كتلة الجمع بالكتلة التالية Machine Parts ومهمتها تصنيع الأجزاء المولدة سابقاً، حيث يستخدم لهذه الكتلة عدد عشوائي بتابع توزع احتمالي منتظم بين القيم 0.65 و 0.7 دقيقة، يبين الشكل التالي النافذة الخاصة بخصائص لهذه الكتلة:



○ الكتلة التالية Inspector Module وهي المسؤولة عن اختبار المنتج بعد التصنيع في الكتلة الثانية Machine Parte، و تتبع هذه الكتلة تابع توزع احتمالي منتظم بين القيم 0.75 و 0.8 دقيقة، يبين الشكل التالي النافذة الخاصة بخصائص هذه الكتلة:



[5 إيضاح] ،

6. الحزم البرمجية غرضية التوجه Application-oriented simulation packeges

31 هناك العديد من الحزم البرمجية والمستخدمة في تطبيقات خاصة نذكر منها:

- نظام AutoMod يستخدم في المصانع [6 إيضاح]
- نظام OPNET يستخدم لمحاكاة شبكات الحاسوبية. [7 إيضاح]
- نظام MedModel يستخدم لمحاكاة المشافي [8 إيضاح]
- نظام Proof Animation يستخدم لمحاكاة الرسوم المتحركة. [9 إيضاح]

ملحقات

[1 إيضاح]

1. برامج جزئية توليد الأعداد العشوائية الزائفة بموزعة بتوزيع منتظم.
2. برامج جزئية توليد الأعداد العشوائية الغير محدد موزعة بتوابع توزيع متعددة.
3. برامج جزئية لتوليد والتحكم بساعة المحاكي.
4. برامج جزئية لتحديد الحدث التالي أو الجديد من قائمة الأحداث المستقبلية.
5. برامج جزئية للتحكم بقائمة الأحداث المستقبلية.
6. برامج جزئية لتجميع وإظهار النتائج.
7. برامج جزئية لدراسة النتائج وتحليلها
8. برامج جزئية لكشف الأخطاء.

[2 إيضاح]

يمكن مثلاً إضافة مخدم إلى نظام محاكاة نظام شبكة عن طريق إدراج أيقونة المخدم ووضعه بالمكان المخصص للمحاكاة ومن ثم ربطه ببقية النظام. ويمكن أيضاً تغيير مواصفات المخدم بالضغط على أيقونة المخدم فتفتح نافذة يمكن من خلالها تغير مواصفات المخدم.

[3 إيضاح]

يمكن للطالب الحصول على مزيد من المعلومات ونسخة عرض Demo من الموقع التالي:

http://www.arenasimulation.com

[4 إيضاح]

يمكن للطالب الحصول على مزيد من المعلومات ونسخة عرض Demo من الموقع التالي:

http://www.extendsim.com

[5 إيضاح]

من شكل الرئيسي للمحاكاة يمكن استنتاج بقية أجزاء المحاكي.

[6 إيضاح]

يمكن للطالب الحصول على مزيد من المعلومات ونسخة عرض Demo من الموقع التالي:

http://www.automedsystems.com

[7 إيضاح]

يمكن للطالب الحصول على مزيد من المعلومات ونسخة عرض Demo من الموقع التالي:

http://www.opnet.com/solutions

[8 إيضاح]

يمكن للطالب الحصول على مزيد من المعلومات ونسخة عرض Demo من الموقع التالي:

http://www.promodel.com/products/medmodel

[9 إيضاح]

يمكن للطالب الحصول على مزيد من المعلومات ونسخة عرض Demo من الموقع التالي:

http://www.wolverinesoftware.com/ProofProducts.htm